Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Detection of COVID-19 Infection Using Chest X-Ray Images Through Transfer Learning and Dense Convolutional Neural Network

HOSAM EL-OCLA [1], (Senior, IEEE), KARTHIK AKSHAY KAMALESH [2], AND RAJEEV SURESH.[3], (Member, IEEE)

[1]Department of Computer Science, Lakehead University
[2]Lakehead University, Thunder Bay, Canada (e-mail: kamaleshk@lakeheadu.ca)
[3]Lakehead University, Thunder Bay, Canada (e-mail: rsuresh1@lakeheadu.ca

Corresponding author: Hosam El-Ocla (e-mail:hosam@lakeheadu.ca).

ABSTRACT An intriguing illness called Covid-19 first surfaced in the Covid family in December 2019, as a result of a recently identified virus. It is an unstoppable illness that mostly affects the human body's lung region and has symptoms similar to those of typical flu, which makes it challenging to recognize. It quickly spread over the world, which has brought up dangerous challenges ever since it started. Such test arrangements should not only be technically solid but also realistic and user-friendly as governments seek to expand testing. Recent X-rays and CT scans have revealed noteworthy highlights that emphasize the severity of Covid in the lungs. Since radiographs like X-rays and CT scans are practical and frequently available at general medical offices, trauma centres, and even in rural hospitals, they could be used for quick identification of potential lung infections brought on by COVID-19. Advanced artificial intelligence (AI) in sending a profound learning-based clinical area is staying fantastic to deal with a huge amount of data with accurate and rapid results in the clinical image to assess illnesses even more precisely and effectively with extra support in far-off places. The proposed model is to use a combination of Deep Learning Models and other classifiers to improve the accuracy of the prediction. The dataset is analyzed and the features are extracted using transfer learning. Sub-dataset is created and the dataset is trained using the different classifiers. It is then tested using a different part of the dataset. Finally, the model can differentiate between covid-19 affected images and healthy images.

INDEX TERMS Convolutional neural networks, COVID-19,X-ray.

I. INTRODUCTION

The course of life has virtually changed in a week. The COVID-19 incident has created a health emergency that has significantly changed how we perceive the environment and daily lives. The Covidflare-up became apparent on December 31, 2019, while China contacted World Health Organization (WHO) about several cases of pneumonia with an enigmatic cause in Wuhan City [1]. After starting in China, it spreads heartbreakingly over the world, causing a tremendous loss of human resources nearly everywhere. Some viruses, including SARS and MARS, are members of the Covid family and have infected a large number of people worldwide [2]. Given its ability to transform, COVID-19 is unique and is anticipated to be the most astounding of all the species in its family. The sickness spread all across the world in this way. Contamination-based treatments necessitate the use of antitoxins, painkillers, or medications. If the patient's condition is complicated, hospitalization is necessary [3] Young children may recover from this illness since they will just experience a mild cold and fewer respiratory problems, but elderly people or those with any chronic illnesses that impairs immunity are at a higher risk. The World Health Organization announced Corona Virus on December 11, 2020, a pandemic that implies a global disease flare-up that might destabilize the entire globe [4].

Respiratory discomfort, fever, and cough are the primary

signs of suspected infections. In cases of more virulent infections, the virus may potentially result in pneumonia. Along with pneumonia, the infection has the potential to cause septic shock, multiple organ failure, severe acute respiratory syndrome, and eventually death [5]. According to studies, males (approximately 60%) were more affected than women (about 40%), and children under the age of nine have not yet experienced any appreciable death rates [2]. Despite being developed, many first-world nations have been experiencing healthcare system collapse as a result of concurrently rising demand for intensive care units [6], [7].

With the global development of new technology, virus tests are becoming faster and faster. A chest scan is used to diagnose COVID-19 infections to confirm the patient's lung state, and if the patient has pneumonia on the scans, a COVID-19 infection is assumed to be present. This technique enables authorities to quickly and effectively segregate and treat affected patients [8].

Despite being open-source data sets, the biggest difficulty that may face Machine Learning (ML) in COVID-19 prediction is the restricted availability of training data sets, as these sets have started with a very small number of images [9], [10]. Researchers should expect the accuracy rate of the trained ML systems to increase as the amount of publicly available data images increases. The majority of artificial intelligence (AI) methods for COVID-19 detection rely on pictures from CT or X-ray scans. However, certain research based on blood testing for COVID-19 screening have been proposed. Gender, age, platelets, basophils, and monocytes are just a few of the variables that have been incorporated into these types of detection models utilizing various AI algorithms, including ANNs, SVM, KNN, decision trees, and random forests [11].

Machine learning methods based on data representation include deep learning and hierarchical learning ideas. The algorithms used by deep learning techniques often use multi-layered neural networks that correspond to different levels of abstraction. The Convolutional Neural Network (CNN), a multi-layer of perceptrons with minimal pre-processing needs, is the most widely used technique in deep learning (DL) for image analysis [12]. The feature extraction procedure can be used to summarise the primary distinction between DL and other machine-learning techniques. In contrast to the conventional technique, which involves the feature engineering process, which is regarded as a costly and difficult procedure [13], the DL network structure is created to extract the crucial information utilizing filters.

Experts are searching for alternate methods to evaluate COVID-19 in influenced people all over the world. As there are no precise automated apparatuses available, the need for assistance indicating devices has increased. Recent research using radiology imaging techniques suggests that these images hold startling information regarding the COVID-19 condition. Additionally, it has been found that using advanced artificial intelligence (AI) algorithms with radiological imaging can help increase the accuracy in identifying this condition and help overcome the problem of a lack of qualified medical staff. This motivated us to develop a mechanism that trains a lot of datasets, uses the latest deep convolutional neural network architectures for binary classification, uses transfer learning for feature extraction and makes the final classifiers robust by utilizing the weights learnt by the pre-trained models.

In this paper, we developed a binary classification system to detect chest X-ray images as COVID-19 patients or healthy patients by the use of deep learning techniques. We used a dense convolutional neural network (DCNN) with ImageNet as the local feature extraction along with the transfer learning method. We used the latest dataset with over 15000 images. The model is validated using 4-fold cross-validation, with the data divided 80:20. A test dataset of 455 photos is used to assess the model's performance, with both classes represented in an equal amount. A strategy that is frequently used to make use of a pre-trained CNN's capabilities is feature extraction via transfer learning. This use of the feature extraction technique along with the transfer learning model reduces the time taken for training and gives you better classification results. The top layers of the Deep Learning architecture will be removed and then a new classifier is trained to conduct classification on top of the previously trained model. The pre-trained model without the final classifier is handled as an arbitrary feature extractor to extract valuable features from the incoming dataset The training set is passed through different classifiers like Densenet, Combination of Densenet and Inception, Combination of Densenet and Resnet, Combination of Densenet and VGG. Finally, the results are measured using metrics like F1 Score, Recall and Precision. The loss evolution and the accuracy evolution are presented in the form of a ROC curve.
In this regard, our contributions in this paper are:

- Developed some models that uses a dense convolutional neural network with ImageNet as the local feature extraction and DenseNet, ResNet, VGG16 and InceptionV3 as the classifier along with the transfer learning method.
  - Processed a dataset that manages to contain more than 15000 images and minimize the training time of these images.
  - Intensity Normalization and Data Augmentation techniques such as rescaling, shifts, rotations shears, and zooms are used to avoid overfitting problems.
- Improve the performance through enhancing the accuracy and minimizing the loss of our models. This is achieved by reducing the pooling layers and simplifying the ReLu activation function.

The structure of this paper is outlined as follows: Section 2 presents a literature review. Section 3 outlines the proposed protocol. Section 4 presents the methodology. Section 5 elaborates on the results. Section 6 summarizes the conclusions.

II. LITERATURE REVIEW

For the past few months, researchers have examined and analysed chest x-ray image samples using a variety of deep learning approaches to look for coronavirus. While some research use augmented and feature extraction approaches, others use raw data. These experiments employed varying numbers of photos. This section discusses a few of these articles.

Recent research comprises the proposal of a deep bayesian ensembling system for the chest CT scan-based programmed detection of COVID-19 cases [14], [15], [16]. To increase the quantity and quality of training data available, data augmentation is used. Informational highlights are separated using transfer learning. The three different bayesian classifiers are created using the extricated highlights. Secured, unrestricted, and regularised Bayesian ensembling approaches are used to evaluate the neural network expectations' susceptibility. The consistency of forecasts is then illustrated. Analyzing the epistemic and aleatoric vulnerabilities and considering several bayesian classifiers from various perspectives. It makes use of a dataset with merely 275 CT images from COVID-19 positive cases. However, this algorithm faces the 'zero-frequency problem' where it assigns zero probability to a categorical variable whose category in the test data set wasn't available in the training dataset.

The authors of [17] apply deep learning to address the COVID-19 detection issue. On pictures from chest CT scans, they employ the EfficientNet model. The plateau strategy performed best on their testing model with an accuracy of 89.7%, followed by the cyclic learning rate which achieved an accuracy of 86%, and a constant learning rate yielded 83% accuracy on their test set. They used fixed, cyclic, as well as learning rate reduced on plateau strategy to fine tune their model. However, in the case of EfficientNets, we have a neural network architecture that has significantly less computing and more data movement than comparable networks. As a result, EfficientNets perform poorly on a larger dataset.

In [18], the authors use the LeNet-5 CNN architecture to chest CT scan images in order to detect COVID-19. Due to the limited availability of CT scan pictures of COVID-19 positive individuals, they applied data augmentation techniques to increase the size of the dataset. Prior to feeding them to the model, all images were converted to grayscale. They used this method, and their accuracy was 86.06%. The authors of [19] detect COVID using low-dose chest CT images. The low-dose CT scan is a safer procedure since it exposes patients to less radiation. They come to the conclusion that the dosage can be decreased by up to 89% while still maintaining the majority of the information after using a deep CNN for the classification procedure. [20] uses a CNN for multiclass classification in which they pool several datasets to amass 10,200 chest CT scans, including normal, COVID-19 positive, and lung cancer patients, then perform conversion to normalise the data. With this approach, they achieved a 90% accuracy rate. However, the main disadvantage of these models is the use of LeNet-5 architecture model, which uses exponential operation for the hidden layer. So, complexity is more for the network. Due to the vanishing gradient problem, there is a loss of data and an increase in the training time.

In [21], the authors suggest a more modern method based on deep learning for COVID-19 recognition from CT-scans. The CT scans are classified by the authors into Normal, COVID-19, or community acquired pneumonia (CAP) classes using CNN architectures with multi-task strategies. The authors separated the CT scans into groups based on the number of slices they included for the patient-level classification stage and used the models they had previously trained on the slices of each group. They succeeded in doing so, with an overall accuracy of 87.75%. In [22], the authors suggest a two-stage CNN-based approach for classifying chest CT scans in order to identify COVID-19 or community-acquired pneumonia. The scientists used the SPGC Dataset, which includes chest CT scans of individuals in the healthy, COVID-19, and CAP classes. Using the DenseNet architecture, the initial stage focuses on identifying a COVID-19 or CAP infection. The authors use a fine-grained multiclass classification with an EfficientNet architecture for the second step. The authors validated their two-stage method by achieving an accuracy of 94% at the slice level classification into COVID-19 or CAP and an accuracy of 89.3% for the three-way classification into COVID-19, CAP, or healthy. However, as you can relate that the above two papers use three way classification which reduces the accuracy to below 90 percent because the machine cannot properly identify between Covid 19 images and Pneumonia images.

In [23], the authors employ transfer learning for 3D Resnet50-based COVID-19 identification. This method uses volumetric analysis for a multiclass classification for the normal, pneumonia, and COVID-positive classes instead of only training on 2D images. Rather than just a single slice from the patients' full volumetric chest CT scans, this information was provided. To extract important information from the slices, they apply an unique preprocessing technique that involves resampling, pulmonary segmentation, and volume range selection. They were 85.56% accurate overall. However, these 3D Resnet50-based COVID-19 identification model has lots of complex networks that don't fit the data making them lose it accuracy.

The authors in paper [24] presents COVID- MobileXpert, a mobile application powered by a lightweight deep neural network. This programme can do Covid-19 Screening on CXR pictures. A lightweight medical student network was used after the deep neural network, which was specifically designed to classify the image as COVID-19 positive or negative, pneumonia or no pneumonia. They deduced from the tests and results that the COVID-19 app built on MobileNetV2 was resource-intensive and had higher accuracy than the COVID-19 app built on SqueezeNet, which was less resource-intensive but had lower accuracy. This indicates that the SqueezeNet is better suited for low-end mobile devices, but the MobileNetV2 based programme could be employed on high performance devices. Even though MobileV2 per-

forms better than the SqueezeNet, it is not as powerful if you train it with larger dataset, the accuracy gets lesser.

The authors of [25] suggest a deep learning-based technique. The model's goal is to identify and locate pneumonia in CXR scans. The localization maps' "Pneumonia Ratio" is a statistic that is used to determine the severity of pneumonia. ResNet50 served as the framework for this deep learning model. The intermediary layers of ResNet were combined by a localization head to create a localization prediction. For detecting pneumonia, a detector head was integrated. For model training, several combinations of the RSNA Pneumonia Detection Challenge dataset, the COVID19 Image Dataset, and the COVID19 Chest Xray Dataset were used. When only the RSNA Pneumonia Detection challenge dataset was employed, the model's accuracy ranged from 0.86 to 0.95, which was the minimum. However, when testing the dataset with localization maps most of the model overfits the data making it lose its accuracy.

SGD optimizers are used by the authors in [26] to get quicker computational performance. Making a real-time COVID-19 detection system was the goal. Using these optimizers, they were able to reduce the amount of time needed for testing and training. This allows for faster classification, which allows for earlier classification of more people and a slower pace of pandemic spread. [27] The CLAHE enhanced model discusses how the current research for Pneumonia diagnosis receives 96% while using VGG16 with transfer learning focuses on only a single modality and only provides about 50% accuracy rate. The authors also compare their CNN using CLAHE against a VGG-16 with transfer learning and show its superior performance data from another modality, such CT scans or X-rays. The authors talk on how several biomarkers could be able to offer data for COVID-19 detection. The authors then suggest combining two transfer learning models, for which they make use of an open-source dataset that has been split into the binary classifications of COVID-19 Pneumonia and Normal. The authors achieve high accuracy by concatenating several transfer learning models, with ResNet50 and VGG16 networks producing the maximum accuracy of 94.87%. However, the dataset that this paper used is very low and thus has more accuracy.

The majority of lung image arrangement is based on a chest x-ray and uses a neural organization model with manual highlights extraction. Recurrent neural networks have been developed for lung image categorization based on chest x-ray scans, and they produce accurate results of 81.33%. The wavelet recurrent neural network model used in this inquiry at the time, which used wavelets for image denoising measures, bears the cost of the outcome with an accuracy of 84%. Wavelet recurrent neural network used the recurrent neural network for the characterization stage and the wavelet to eliminate picture commotion in the preprocessing step. The manual highlights extraction measure used in these earlier studies was a contribution to the categorization cycle [28], [29]. However, using Recurrent neural networks the computation of the network is slow, and it faced issues like Exploding or Gradient Vanishing Problems.

Numerous scientists from various foundations [28] are actively interested in developing strong diagnostic tools and vaccines to combat the pandemic flare-up. The analysts' area contains experts from various topics, such as data science, machine learning, and deep learning, in addition to the medical and biotechnology sectors. The use of X-rays is dependent on how once Covid reaches the respiratory tract, it will affect the patient's lungs and result in pneumonia. In this case, the lungs fill up with liquid, ignite, and produce patches known as "Ground-Glass Opacity." This enables the detection of COVID-19 using just the chest X-rays of infected individuals. It is impressive that X-ray equipment can scan a variety of human organs. X-rays are routinely translated for patients by master radiologists. However, many cases of covid 19 detection cannot be found using "Ground-Glass Opacity" because covid 19 has many varieties and it differs for each variant.

Narin et al. [30] suggested combining X-ray pictures and deep learning to predict Covid19 automatically (2020). Three Deep Convolution Neural Network designs were used in the suggested strategy. They used a dataset with 50 normal X-ray pictures and 50 images of covid19 patients, all of which were scaled to 224 * 224. The authors used transfer learning models to get around the issue of the small dataset. Eighty percent of the dataset was used for training, while twenty percent was used for testing. The created DCNN was based on pre-trained models that could recognize Covid19 from standard X-ray pictures (ResNet50, Inception V3, and Inception ResNet V2). They also employed a transfer learning strategy, and they used the k-fold method in conjunction with k-fold to cross-validate their results. However, the dataset that they used has only 50 images and even though they got better results, we cannot use them for real life because the model does not learn enough when you have less dataset.

A recent study [31] used an automated covid detection technique to categorize the X-ray pictures into three groups in order to determine the covid positive (covid positive, viral pneumonia, normal cases). Custom CNN, pretrained VGG16, and pretrained ResNet50 were each employed independently for the task. The chest X-ray pictures were obtained using a regular dataset from Kaggle (219 covid positive samples, 1345 viral pneumonia samples, 1341 normal samples). The effectiveness of the model using each of the three ways was evaluated. The customised CNN model produced results with an accuracy of 93.67%, precision of 95.65%, recall of 94.66%, and F1 score of 95.03%. VGG16 produced results with accuracy of 97.67%, precision of 96.65%, recall of 96.54%, and F1 score of 96.59%. Resnet50 model produced results with accuracy of 94.41%, precision of 93.58%, recall of 93.21%, and F1 score of 93.29%. The pretrained VGG16 outperformed the bespoke CNN and Resnet50 models by a significant margin. For the purpose of coronavirus detection from x-ray pictures, another study [32] offered 5 pre-trained models based on deep CNNs (Resnet50, Resnet101, Resnet152, InceptionV3, InceptionResnetV2).

Three datasets were utilized in the experiment: 1493 viral and 2772 bacterial pneumonia image samples from the Kaggle repository, 341 covid positive image samples from dataset, 2800 normal image samples from "ChestX-ray8" dataset, and 341 covid positive image samples from dataset. Additionally, three unique datasets were made utilising these (dataset1: covid and normal cases; dataset2: covid and viral pneumonia cases; dataset3: covid and bacterial pneumonia cases). The model used fivefold cross validation, and the accuracy results were as follows: InceptionV3 (dataset1- 95.4%, dataset2- 98.6%, dataset3- 97.7%), InceptionResnetV2 (dataset1- 94.2%, dataset2- 94.4%, dataset3- 95.3%), Resnet50 (dataset1- 96.1%, dataset2- 99.5%, dataset3- 99.7%), Resnet101 (dataset1- 96.1%, dataset2- 93.5%, dataset3- 94.7%), In this situation, pretrained Resnet50 outperformed other pretrained models. However, the main problem with these techniques is the use of pretrained models. Each paper gets different model as the best model accuracy and this is because of the dataset and Many pre-trained models are trained for less or more different purposes ,so may not be suitable in some cases.

III. PROPOSED PROTOCOL

A. PROBLEM STATEMENT

Covid 19 Detection using Chest X-Ray images uses fewer and imbalanced datasets for Classification. They use Grayscale and Histogram of Oriented Gradients (HOG) as feature extraction along with pre-existing classifiers such as Support Vector Machine (SVM), Naive Bayes Classifier, K-Nearest Neighbour (KNN), and Decision Tree for classification. The major problem in all these previous papers and their techniques is that the lack of dataset to train these images. Also all the machine learning classifiers has some kind of problems like the gradient vanishing problem, underfitting and overfitting the bias problems. It is desired to propose a protocol that considers these problems and decreases the loss percentage of the training model and therefore enhances its performance.

B. PROPOSED SOLUTION

In this paper, we present our deep learning models that uses Dense Convolutional Neural Network with ImageNet as Feature Extractor and use transfer learning method for Classification. We merged all the datasets available online and made it a larger Dataset. Then to make the dataset balanced we use Data Augmentation techniques like Rescaling, ZCA whitening, Fill Mode, Holizonatal Flip, and Rotation. A pre-trained CNN model that was previously trained with millions of images is transferred in the feature extraction process using deep transfer learning. The pre-trained model's final layer, which is utilized for classification, will be dropped, though. Then, a new classifier is trained to conduct classification on top of the previously trained model. The pre-trained model without the final classifier is handled as an arbitrary feature extractor to extract valuable features from the incoming dataset. The network design of the CNN model is changed to exclude the task-dependent levels, such as the fully-connected layers and the classification output layer while keeping the other layers for the new classification task. Increasing the Number of Epochs and reducing the network layers reduce the model's training time.

IV. METHODOLOGY

This section outlines the suggested methods for identifying if a patient on an X-ray is healthy or has COVID-19. We'll start by outlining the image datasets that were employed in this investigation. The feature extraction procedure, which is based on the transfer learning theory, is next explained. The steps of their training are then presented, along with the categorization techniques used. Finally, we outline the metrics we use to assess the outcomes and contrast them with those of alternative strategies. Figure 1 shows the architecture of the proposed methodology.

A. DATASETS

We employ frontal-view chest X-ray pictures in our research. Only the anterior-posterior (AP) and posterior-anterior (PA) images of the X-ray were gathered. We separated the samples into two groups: COVID-19 patients with X-ray images and healthy patients without COVID-19 diagnoses. We created three datasets, Dataset A and Dataset B, and Dataset C to better assess the suggested strategy. The COVID-19 class has the same photographs in both datasets, whereas the healthy class has different images.

We employ frontal-view chest X-ray pictures in our research. Only the anterior-posterior (AP) and posterior-anterior (PA) images of X-rays were gathered. We separated the samples into two groups: COVID-19 patients with X-ray images and healthy patients without COVID-19 diagnoses. We created three datasets to better assess the suggested method: Dataset A contains the training images which has 15,370 images, but Dataset B and Dataset C are used for testing purpose only.

The 15,360 chest X-ray images of patients who have been diagnosed with COVID-19 that make up the COVID-19 class in Dataset A were gathered from various sources [33], [34]. 15,370 photos make up the collection, of which 6276 belong to the COVID-19 class and 9094 to the Normal class. These two sources were retrieved on March 31, 2020. They are collections of X-ray pictures gathered from various publications, databases, and other sources.

The 300 chest X-ray images of COVID-19-diagnosed patients in Dataset B, which were gathered from several sources [35], [36], make up the COVID-19 class. To represent healthy patients, we randomly chose 150 samples from the X-ray scans marked as "normal." This source was chosen because it is frequently cited in publications that are related to this one and suggest techniques for identifying COVID-19 in X-rays. These images were used for testing purposes.

As indicated earlier, Dataset C has the identical COVID-19 class photos as Dataset A. However, for Dataset C, we gathered chest X-ray pictures from a different source due to

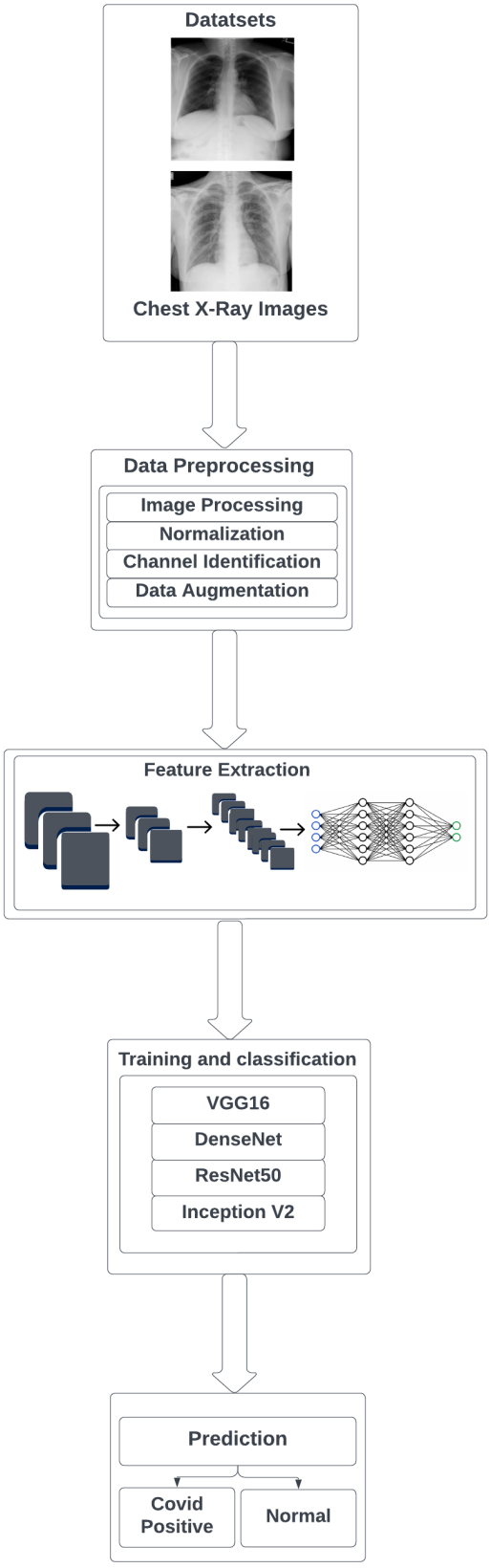

FIGURE 1: System Overview Diagram.

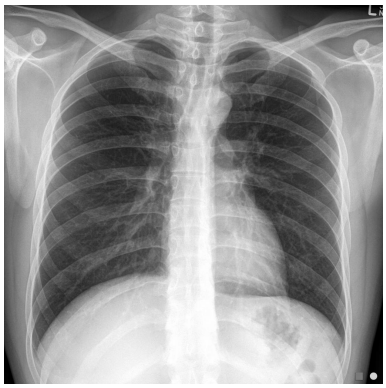

FIGURE 2: Normal X-Ray

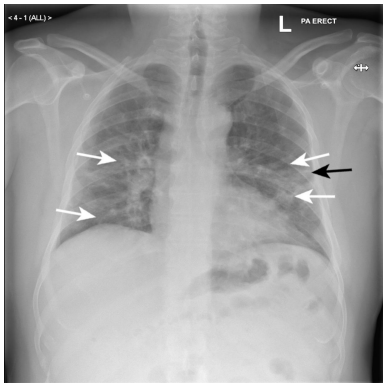

FIGURE 3: Covid-19 X-Ray

the age difference between healthy patients and those with COVID-19 seen in Dataset A. The National Institutes of Health hosted the "NIH Chest X-rays" competition, which was made available on Kaggle [37], and we used those photos for this dataset. 194 photos representing healthy patients in the "no finds" class were randomly chosen.

The datasets' photos are all either in the portable network graphics (PNG) or the joint photographic experts group (JPG/JPEG) format. There are different image resolutions in the collection, ranging from 249 by 255 pixels to 3520 by 4280 pixels. In contrast to Figure 3, which shows lungs with a more diffuse "interstitial" pattern, Figure 2 shows clean lungs without any aberrant opacification zones.

B. DATA PREPROCESSING

By removing or lowering the noise that was present in the original input image, increasing contrast, removing low or high frequencies, and other techniques, image pre-processing is used to enhance the visual information of each input image. The input photos are preprocessed at this point utilizing a variety of pre-processing techniques (Table 1). Intensity normalization, a pre-processing phase in image processing software, is used for this project. The formula below demonstrates how min-max normalization is used in our research to transform input photos into the conventional normal distribu-

tion.

$$X_n = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

where X_n is the normal function processing each image, X is the image size in terms of pixels, X_{min} is the minimum intensity of the image, and X_{max} is the maximum intensity of the image.

C. DATA AUGMENTATION

After splitting and pre-processing the dataset, data augmentation is carried out to prevent overfitting. To do this, geometric transformations like rescaling, rotations, shifts, shears, zooms, and flips are used. All of the data augmentation strategies employed are listed in the table below.

D. FEATURE EXTRACTION

We employ the transfer learning idea to extract characteristics from the X-ray images. First, we choose various CNN designs that performed exceptionally well on the ImageNet dataset. Second, we select various CNN architectures from the list that have been previously trained on ImageNet. Thirdly, all fully connected layers are eliminated from these setups, leaving only convolutional and pooling layers. The completely linked layers are in charge of classifying the features and, as a result, the image, while the other two types of layers are in charge of extracting features from the image. Therefore, to convert a CNN into a feature extractor, these layers must be removed. Following this phase, the modified CNN's new output is a collection of features retrieved from the input image.

We produce a sub-dataset for each CNN configuration that consists of sets of traits that have been taken from each of the unique datasets. To create a sub-dataset, we first resize the original data. each picture in accordance with the input size needed by the chosen CNN. Each image is then utilized as input to the CNN, and the features it contains are extracted, then saved in the equivalent sub-dataset. The Architecture of the feature extraction using dense convolutional neural network is shown in Figure 4.

A pre-trained CNN model that was previously trained with millions of images is transferred in the feature extraction process using deep transfer learning. The pre-trained model's final layer, which is utilized for classification, will be dropped, though. Then, a new classifier is trained to conduct classification on top of the previously trained model. The pre-trained model without the final classifier is handled as an arbitrary feature extractor to extract valuable features from the incoming dataset. The network design of the CNN model is changed to exclude the task-dependent levels, such as the fully-connected layers and the classification output layer, while keeping the other layers for the new classification task.

We are using an input image at this time. Convolution + pooling processes are carried out repeatedly, then a number of completely linked layers are added as shown in Figure 5. The convolutional layer is the foundation of CNN. A

Argument	Parameter value	Description
Rescale	1 / 255.0	Scale images from integers 0-255 to floats 0-1
Rotation range	10	Degree range of the random rotations
Width shift range	0.2	The parameter value of horizontal shifts (20%) in a fraction of the give dimension
Height shift range	0.2	The parameter value of vertical shifts (20%) in a fraction of the given dimension
ZCA whitening	True	Reduces the redundancy in the matrix of pixel images and keeps all of the original dimensions, unlike PCA (Principal Component Analysis)
Fill mode	Nearest	The closest pixel value is chosen and repeated for all the empty values
Horizontal flip	True	Controls when a given input is allowed to be flipped horizontally during the training process
Zoom range	0.1	Allows the image to be "zoomed out" or "zoomed in"

TABLE 1: Data Augmentation Techniques

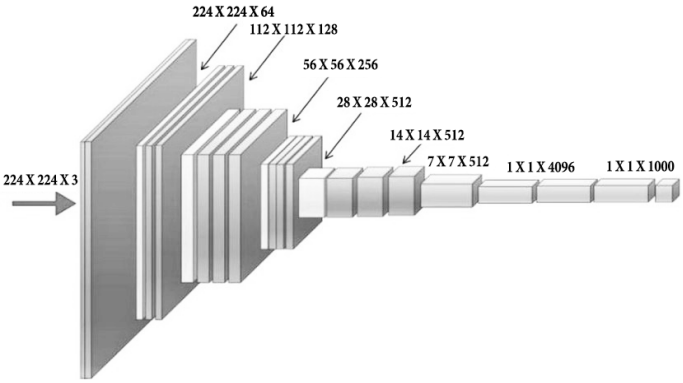

FIGURE 4: Feature Extraction Architecture of X-Ray Image

mathematical procedure called convolution is used to combine two sources of data. In our case, a convolution filter is used to apply convolution to the input data in order to create a feature map. We apply this filter to the input before doing the convolution procedure. We perform element-wise matrix multiplication at each point and add the outcome. This amount is included in the feature map. The actual representation of an image is a 3D matrix with the parameters height, width, and depth, where depth corresponds to colour channels (RGB). A convolution filter must be 3D because by design it covers the entire depth of its input and has a fixed height and breadth.

We typically employ pooling to lower the dimensionality after a convolution operation. We can do this to cut down on the number of parameters, which reduces training time and prevents overfitting. By combining layers, each feature map is independently downscaled, preserving the depth while lowering the height and breadth.A few fully connected layers are added to the CNN architecture after the convolution + pooling layers.Convolution and pooling layers both provide

Configurations	Input image size (pixels)	Number of features extracted
VGG16	224 x 224	512
ResNet50	224 x 224	2048
DenseNet201	224 x 224	1920
InceptionV3	299 x 299	2048

TABLE 2: CNNs Configurations, Their Input Image Size and Number of Features Extracted

3D volumes as their output, while a fully linked layer anticipates a 1D vector of integers. Therefore, the last pooling layer's output is flattened to a vector, which is then used as the input for the fully connected layer. Simply arranging the 3D volume of integers into a 1D vector is flattening.

It is difficult to automatically identify significant features from just an image and a label. The convolution layers build up on one another to understand such intricate details. The first layers identify edges; the second levels combine them to identify forms; and the third layers combine this data to determine that this is a lung image. It gains the ability to recognize something as a feature by repeatedly viewing them in images. Convolutional features are taught to the fully linked layers so they can accurately identify the images.

The simple concept of dropout is utilized to avoid overfitting. Every repetition during training period results ,in the temporary "dropping" or "disabling" of a neuron with probability p. This indicates that at the current iteration, all of this neuron's inputs and outputs will be disabled. At every training step, the dropped-out neurons are resampled with probability p, allowing a dropped-out neuron at one step to be active at the following one. The dropout-rate, or hyperparameter p, is typically about 0.5, or 50% of the neurons, and it is expressed as a number.

Each CNN model that we were using has a different configuration and a different number of features extracted. Table 2 shows all the different configurations and features extracted. We can see that VGG16 has the lowest features extracted because it has 5 Convolutional layers and 5 block layers used. The total parameters that we trained are 21,137,729, whereas Trainable params are 6,423,041 and Non-trainable params are 14,714,688.

The ResNet50 Architecture has around 2048 features extracted and their input size is 299 x 299 pixels. We used around 48 convolutional layers, one MaxPool layer and one average pool layer. The total parameters are 23,564,800 whereas Trainable params are 0 and Non-trainable params are 23,564,800.

The Densenet201 has around 200 layers. we used 5 Convolutional layers, 38 block layers for the architecture. The total parameters trained are 42,406,977 whereas the trainable params are 24,084,993 and the non-trainable params are 18,321,984.

The InceptonV3 network has 48 layers deep. We used around 7 Convolutional layers and 14 Conv2D layers for this network. The total parameters used are 34,910,497 whereas trainable parameters are 13,107,713 and Non-Trainable pa-

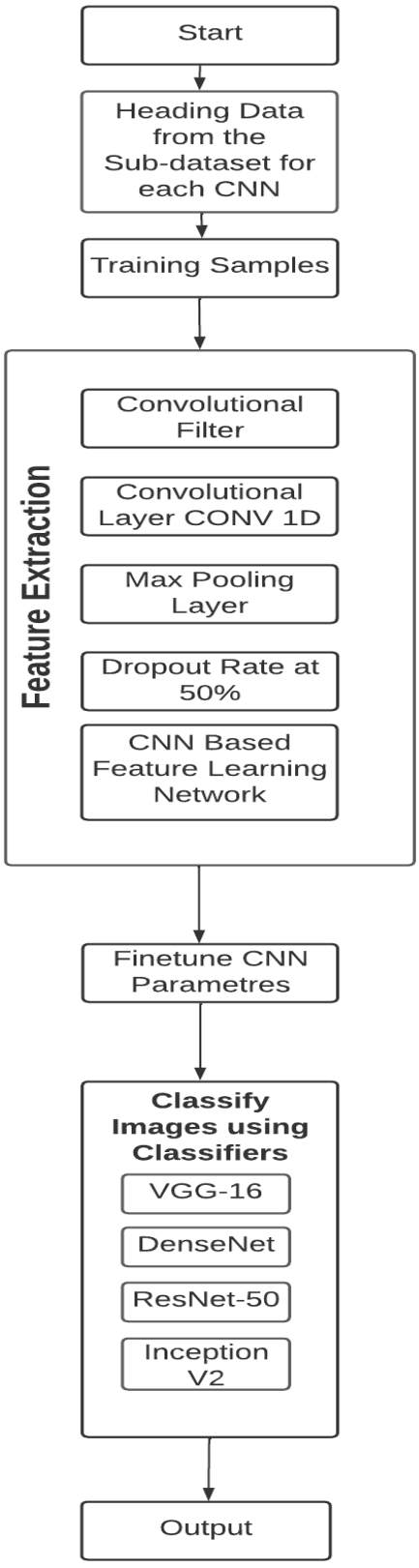

FIGURE 5: Flow chart of Feature Extraction Method using CNN.

rameters are 21,802,784.

E. CNN ARCHITECTURE

The pre-trained models of the CNN Architecture are trained from beginning to end to enable feature extraction and feature selection, followed by classification and prediction. CNNs' spatial filters automatically pick up information about the structure in the picture [38]. In contrast to conventional machine learning image classification methods, CNN's operate pixel-by-pixel on pictures [38]. The input layer, convolutional layers, pooling layers, reLU layers, and inner-product layers, or fully connected layers, make up the five layers that make up the CNN architecture. X-ray pictures will be utilized as inputs in the Input layer, with the parameters setting the image dimension (244 x 244). Multiple convolutional and pooling layers are used by CNN to monitor and assess possible features. In the convolutional layers, a two-dimensional array of weights (filters) are multiplied. In order to lower the spatial size of convolutional features and address overfitting issues, the max-pooling layer is utilized. It extracts the largest area found in the preceding convolutional layer. The maximum value in each patch is found using the Max pooling approach. Rectified linear unit (ReLU) is an activation function that maps between an input and its target variable using a linear function. Therefore, the pre-trained model's performance Fully connected layers are employed as a classifier that categorizes things based on characteristics. Inner-product layers or fully linked layers are used to accept a straightforward vector as input and produce a single vector as output. These layers use a sigmoid activation function.

F. DEEP LEARNING ARCHITECTURE

Training increasingly complex neural networks takes more labour. End-to-end multilayer integration of low, mid, and high level features and classifiers occurs naturally in deep networks. Additionally, the amount of stacked layers can raise the depth or feature levels. In our study, deep feature extraction and fine-tuning methods are carried out using the VGG16, ResNet50, Densenet201, and Inception V3 models.

1) VGG

VGG, or Visual Geometry Group, is an architecture for a convolution neural network (CNN). The key distinction between this design and others is that it emphasizes straightforward 22 size kernels in max-pooling layers and simple 33-size kernels in convolutional layers. It then follows with a softmax and two Fully Connected output layers. After the last fully connected layer, the model uses a softmax classifier and max-pooling to decrease volume size. The VGG16 input is an RGB picture with a fixed size of 224 by 224. It consists of 16 layers, of which 3 are completely linked and 13 are convolutional. Convolutional neural networks have been successfully used as the fundamental network in image recognition techniques [39] .

2) ResNet50

The ResNet50 model, which was suggested in "Deep Residual Learning for Image Recognition" [40] will be employed in this project. Resnet50 is a deep residual network that is used for image classification and is a member of the convolutional neural network model family. A novel network-in-network design built on residual layers is the key innovation. Five stages make up the Renset50, each of which has a convolution and identity block with three convolution layers. It has one MaxPool layer, one average pool layer, and 48 convolutional layers. Resnet50 can train more than 23 million parameters and contains 50 residual networks. Images having a resolution of 224 by 224 pixels are acceptable. When deep networks are being trained, accuracy reaches a saturation threshold at which time it rapidly deteriorates. This issue is referred to as "degradation" This demonstrates how not every neural network architecture is simple to optimise. ResNets may be used to solve a variety of issues. The accuracy of these systems may be improved and they are simple to optimise.

3) DenseNet201

This 201-layer deep convolutional neural network is called DenseNet201 (Dense Convolutional Network). A ResNet enhancement known as DenseNet201 has thick connections between layers. Compared to conventional convolutional networks with L layers and L connections, DensNet201 has $L(L+1)/2$ direct connections. It links every layer to every other layer in a feed-forward manner. By requiring more computing, using fewer parameters, promoting feature reuse, and bolstering feature propagation, DenseNet can outperform conventional networks. It will take images up to 224 224 in size. DenseNet was created primarily to enhance the decreased accuracy brought on by high-level neural networks' vanishing gradient. In plainer terms, the information disappears before getting there because of the lengthier journey between the input layer and the output layer. Due to the new architecture of dense connectivity, which enables interconnection between arbitrary layers, skip connection mode, which transmits data from shallow layers directly to deep layers, and enhanced feature transfer and feature reuse between network layers, DenseNet [41] has outperformed many other deep learning models in terms of performance. This leads to a more compact network representation with less feature redundancy. Similar layer depth improves network convergence performance, mitigates network degradation and gradient disappearance issues brought on by convolutional network deepening, and considerably reduces the number of network parameters and computing efficiency

4) InceptionV3

The most recent iteration of the Inception model, Inception V3 is an enhanced version of Inception V1 and V2. There are 484 layers and 11 conception modules in it. Each module includes the ReLu activation function, pooling layers, and

convolution filters. By factorising convolutions, Inception V3 lowers the number of parameters while maintaining network effectiveness. The inception model has a different structure from the standard CNN model since the models are inception blocks. There are three possible filter sizes in a block of parallel convolutional layers (1x1, 3x3, 5x5). Additionally, 33 max-pooling is carried out. The inception model concatenates the results after lapping the same input tensor several times with various filters [42]. On the ImageNet dataset, it has been demonstrated that the image recognition model Inception v3 achieves higher than accuracy. The model is the result of several concepts that have been established by various scholars throughout the years. The primary goal of Inception v3 is to consume less computing resources by altering the Inception structures from earlier versions. Inception Networks have shown to be more computationally efficient in terms of both the amount of parameters they provide and the cost-effectiveness of their implementation (memory and other resources). It is important to take care not to lose the computational benefits while making changes to an Inception Network. Due to the unknown effectiveness of the new network, it becomes difficult to modify an Inception network for various use cases. In an Inception v3 model, a number of network optimization strategies have been proposed to relax the restrictions and make model adaption simpler. The methods include regularisation, dimension reduction, factorised convolutions, and parallelized calculations.

G. EVALUATION METRICS

The ultimate purpose of employing evaluation metrics is to determine how well a machine learning model will perform when dealing with unknown data [17], [30].

1) Accuracy

The number of correct predictions divided by the total number of input samples is known as accuracy. It only works when there are an equal number of samples in each class.

$$Accuracy = \frac{NCP}{TPS} \tag{2}$$

where NCP represents the Number Of Correct Predictions and TPS denotes Total Predictions

2) Precision

The number of correct positive results divided by the number of positive results predicted by the classifier is what is known as precision.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

where TP denotes True Positives and FP represents False Positives.

3) Recall

The number of correct positive results divided by the total number of relevant samples (all samples that should have been detected as positive) is known as recall.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

where TP denotes True Positives and FN represents False Negatives.

4) f1-score

The weighted average of Precision and Recall is the f1 Score. As a result, this score takes into account both false positives and false negatives. When the data is unbalanced, this method is employed.

$$\text{f1-score} = 2 * \frac{(RC * PN)}{(RC + PN)} \tag{5}$$

where RC represents Recall and PN specifies Precision.

V. EXPERIMENTS AND RESULTS

In our experiment, a publically accessible image dataset was subjected to binary classification using the following parameters: To train our models, we employed a batch size of 32 and a total of 50 epochs, while Adam is used for optimization with a learning rate of 0.001. The dataset's photos were all downsized to 224x224 pixels.

The produced models are run on a computer running Microsoft Windows 10 Professional with an AMD (Ryzen) 5800h CPU clocked at 3.60 GHz and 16 GB of RAM (64-bit). Keras/TensorFlow is used on the backend. The NVIDIA GTX 1660ti with 8 GB V-RAM is used to execute our validation and training operations.

The dataset was trained using four-fold cross-validation with a 75:25 data split across 25 iterations, with a batch size of 10. The training method also included the usage of a 75:25 data split. The tabulated and reported findings are presented in Tables 2a, 2b, 2c, and 2d. The complete dataset consisted of 14,915 photos from the Normal and Covid classes. The Normal class contained 8,767 photos, while the Covid class contained 6,148 images. The chest X-Ray dataset's binary classification results, which were anticipated from experiments learned on the training set, are shown on the test set. Using the VGG16, ResNet50, DenseNet201, and InceptionV3 architectures, we assessed the deep neural network properties.

The four models' performance on a set of test data for which the true values are known is given below in the confusion matrix for each of the four models.

A. CONFUSION MATRIX

1) Model VGG16

The model correctly predicted 239 photos in the Covid class, but 17 of these were incorrectly classified as Normal, as seen in the confusion matrix in Figure 6. 283 of the photos in the Normal class were correctly categorised by the model. No photos were wrongly classified as belonging to the Covid class, hence there were also no false positives.

	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
Fold 1	98.45	3.8	98.89	4.25
Fold 2	98.65	3.95	98.49	3.95
Fold 3	98.13	4.01	99.53	2.70
Fold 4	97.78	3.66	97.21	3.6
Average	98.25	3.85	98.53	3.6

Table 2a. VGG16 Training Evaluation

	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
Fold 1	99.28	2.2	99.22	1.8
Fold 2	98.88	2.4	99.12	1.6
Fold 3	99.3	2.1	99.8	0.9
Fold 4	97.1	2.7	97.23	1.74
Average	98.64	2.35	98.84	1.51

Table 2b. ResNet50 Training Evaluation

	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
Fold 1	99.3	1.5	99.4	1.7
Fold 2	99.2	1.2	98.6	1.2
Fold 3	98.7	1.6	97.9	1.9
Fold 4	98.9	2.1	99.2	2.3
Average	99.02	1.6	98.77	1.8

Table 2c. DenseNet201 Training Evaluation

	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
Fold 1	98.48	4.07	99.09	2.1
Fold 2	98.62	3.6	98.72	3.7
Fold 3	98.3	4.4	99.12	2.3
Fold 4	98.1	3.1	97.89	5.1
Average	98.37	3.8	98.7	3.3

Table 2d. InceptionV3 Training Evaluation

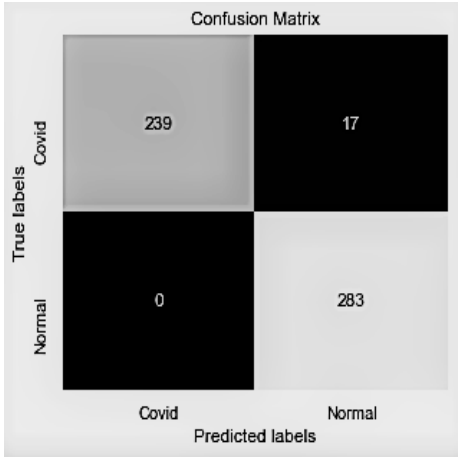

FIGURE 6: Confusion Matrix(VGG16)

2) Model ResNet50

According to the confusion matrix Figure 7, the model properly assigned 283 images to the Normal class while misclassifying 0 images as Covid for the Normal class. Using the ResNet50 model, 12 images were labelled as Normal and 244 images were correctly predicted to belong to the Covid class.

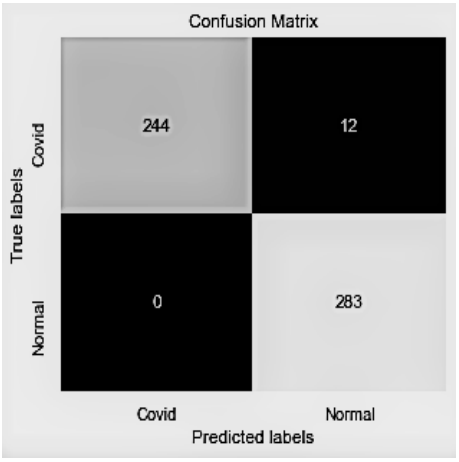

FIGURE 7: Confusion Matrix(ResNet50)

3) Model DenseNet201

The model correctly predicted 283 photos in the Normal class, but 7 of these were incorrectly labelled as Covid, according to the confusion matrix in Figure 8. The algorithm also properly predicted 248 images for the Covid class, with 0 images incorrectly classified as Normal.

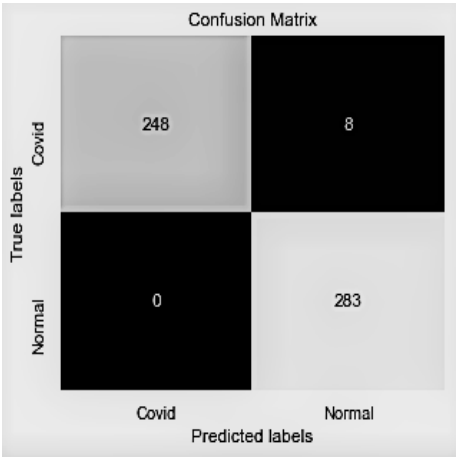

FIGURE 8: Confusion Matrix(DenseNet201)

4) Model InceptionV3

According to the confusion matrix in Figure 9, the model correctly identified 276 pictures as belonging to the Normal class, but misclassified seven others as Covid. And, while 3 images were incorrectly classified as belonging to the Normal

class, the algorithm correctly predicted 253 images for the Covid class.

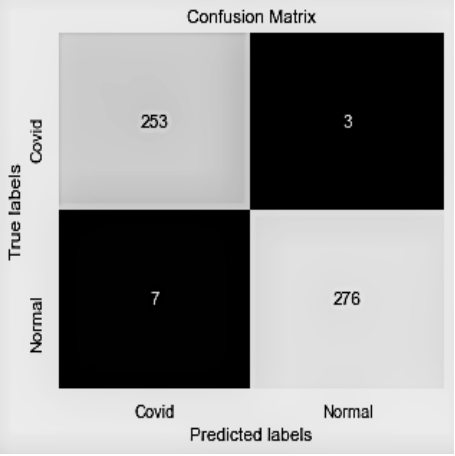

FIGURE 9: Confusion Matrix(InceptionV3)

B. PRECISION, RECALL AND F1-SCORE

Figures 10,11,12 and 13 below display each model's precision, recall, and f1-score. It provides more detailed information about the classification outcomes of each trial.

	Covid	Normal	accuracy	macro avg	weighted avg
precision	1.000000	0.943333	0.96846	0.971667	0.970247
recall	0.933594	1.000000	0.96846	0.966797	0.968460
f1-score	0.965657	0.970840	0.96846	0.968249	0.968378

FIGURE 10: Precision, Recall and f1-Score(VGG16))

	Covid	Normal	accuracy	macro avg	weighted avg
precision	1.000000	0.959322	0.977737	0.979661	0.978642
recall	0.953125	1.000000	0.977737	0.976562	0.977737
f1-score	0.976000	0.979239	0.977737	0.977619	0.977700

FIGURE 11: Precision, Recall and f1-Score(ResNet50)

	Covid	Normal	accuracy	macro avg	weighted avg
precision	1.000000	0.972509	0.985158	0.986254	0.985566
recall	0.968750	1.000000	0.985158	0.984375	0.985158
f1-score	0.984127	0.986063	0.985158	0.985095	0.985143

FIGURE 12: Precision, Recall and f1-Score(DenseNet201)

C. ACCURACY AND LOSS GRAPHS

1) Training Evaluation For Dataset A

The figures below depict the accuracy and loss curves during the training and validation processes for each model. The resulting train data accuracy curve for the DenseNet201

	Covid	Normal	accuracy	macro avg	weighted avg
precision	0.973077	0.989247	0.981447	0.981162	0.981567
recall	0.988281	0.975265	0.981447	0.981773	0.981447
f1-score	0.980620	0.982206	0.981447	0.981413	0.981453

FIGURE 13: Precision, Recall and f1-Score(InceptionV3)

model starts to stabilise at 98 percent accuracy. Up to epoch 2, the accuracy grew steadily, reaching 97.5%. Figure 14 and Figure 15 shows that the train and validation data loss curves were largely constant at 2.5%.

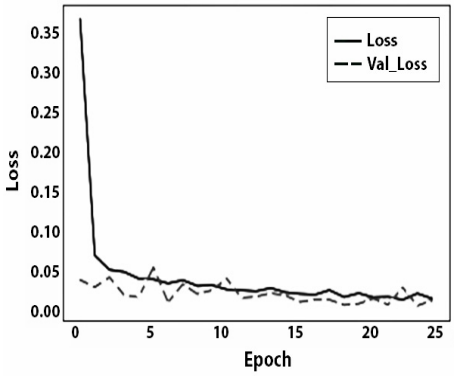

FIGURE 14: Loss curve for Training and Validation (DenseNet201) For Dataset A

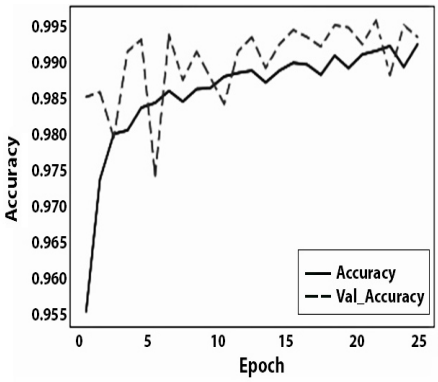

FIGURE 15: Accuracy curve for Training and Validation (DenseNet201) For Dataset A

The accuracy for the train data for the ResNet50 model quickly rises to 97.7% between epochs 1 and 2 (Figure 16). Then, after epoch 10, it achieves stability. Validation accuracy generally stayed consistent at around 99%. At epoch 1, the training data loss curve (Figure 17) drops significantly to 6.5% before stabilising at epoch 5. The validation data's loss curve stayed at 2% the entire time.

The Inception V3 model's accuracy curve, shown in Figure, is stable starting at epoch 5 for both train and validation accuracy. Prior to that, the train data accuracy curve increased continuously until epoch 1, when it reached 96.9%. Figure 18

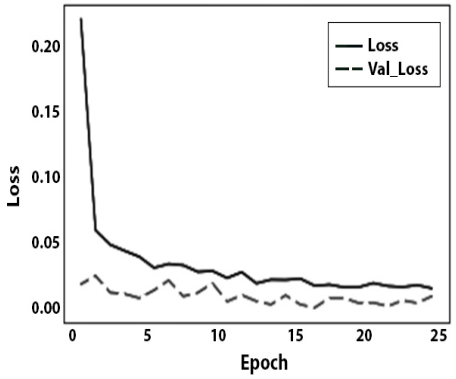

FIGURE 16: Loss curve for Training and Validation (ResNet50) For Dataset A

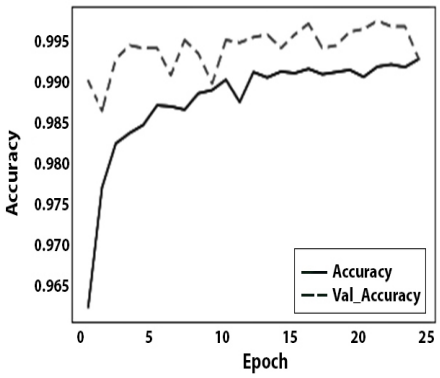

FIGURE 17: Accuracy curve for Training and Validation (ResNet50) For Dataset A

and Figure 19, which depicts the loss curve, demonstrates a continuous loss of 5% for the train and validation data after epoch 1.

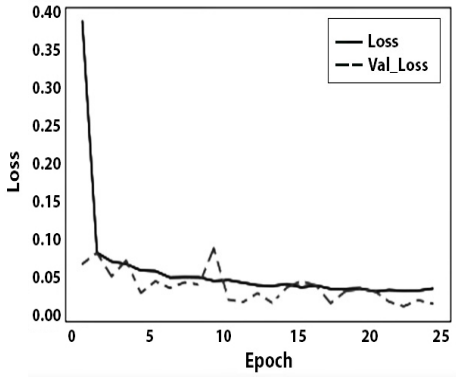

FIGURE 18: Loss curve for Training and Validation (InceptionV3) For Dataset A

With a 96.8% accuracy, the train data accuracy curve for the VGG16 model rapidly increases between epochs 1 and 2. After that, at epoch 5, the train data accuracy curve converges to a point of 98 percent accuracy before stabilising. The train data loss curve, shown in Figure 20 and Figure 21, similarly

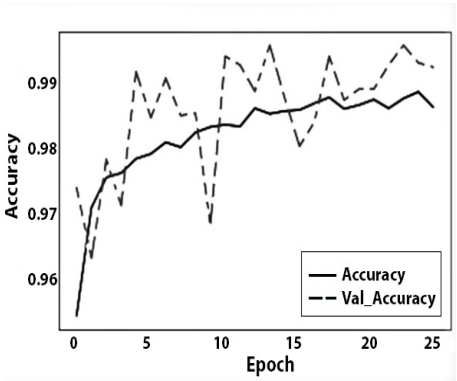

FIGURE 19: Accuracy curve for Training and Validation (InceptionV3) For Dataset A

exhibits a sharp decline between epochs 1 and 2, when the loss was roughly 9%, followed by a gradual decline until epoch 5, where the loss was 6%, before stabilizing.

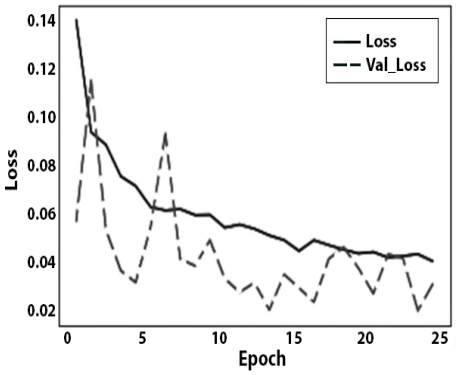

FIGURE 20: Loss curve for Training and Validation (VGG16) For Dataset A

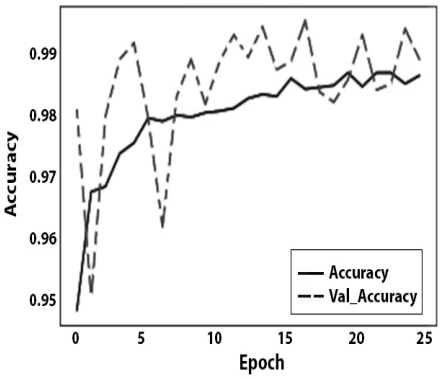

FIGURE 21: Accuracy curve for Training and Validation (VGG16) For Dataset A

2) Testing Evaluation For Dataset B

In Figure 22, it is visible that all the models have more than 98% accuracy. We can see that DenseNet has the least devi-

TABLE 3: Accuracy and Loss on Test Dataset A

Models	Testing Accuracy	Testing Loss
DenseNet201	98.51	4.8
InceptionV3	98.1	8.8
ResNet50	97.7	9.2
VGG16	96.84	10.7

ation and touches the maximum percentage of 99.5 percent. ResNet contains the 2nd most maximum curve but contains sharp deviation in accuracy. VGG16 and InceptionNet contain the least curve among the 4 models and get an average of 98 percent accuracy.

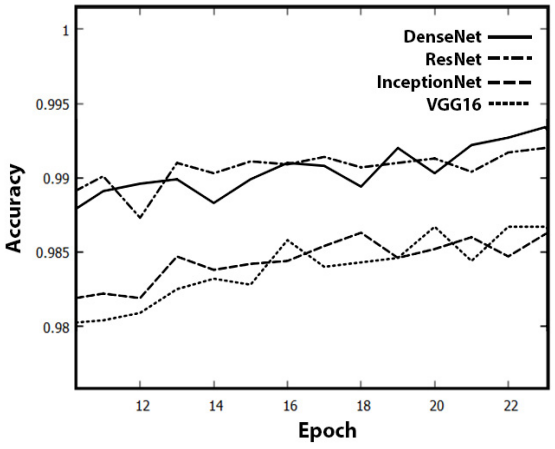

FIGURE 22: Comparison of Training accuracy between the four models for Dataset B

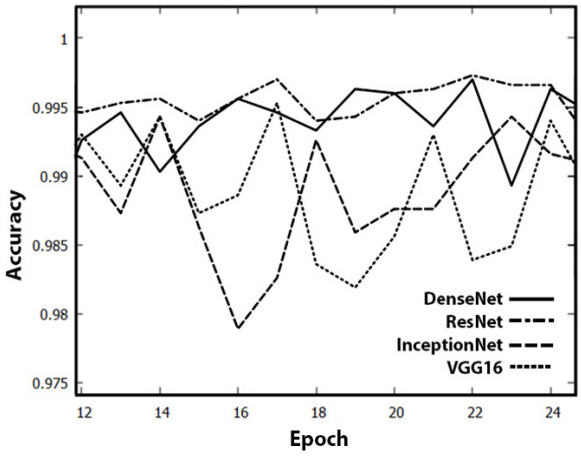

FIGURE 23: Comparison of Validation Accuracy with the four of our Models for Dataset B

In Figure 23, it is visible that the DenseNet curve and ResNet curve has the least deviation compared to other two curves. VGG16 in epoch 16 gets the minimum curve whereas ResNet in epoch 17 gets the maximum peak accuracy. Thus we can conclude that DenseNet and ResNet perform the most.

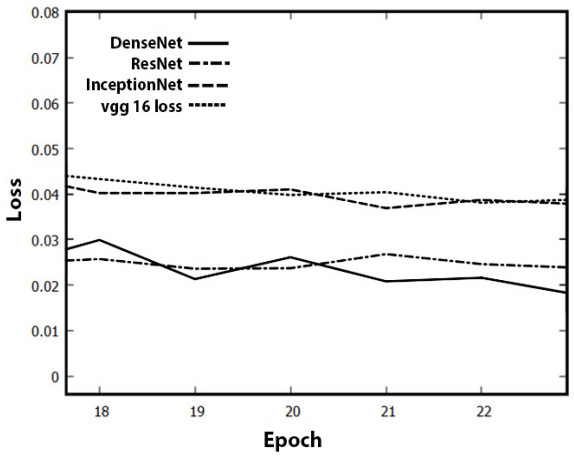

FIGURE 24: Comparison of Training Loss with the four of our Models for Dataset B

In Figure 24, we compare the loss and loss validation of the four models. We can see that loss of DenseNet touches the lowest point of 0.02 in epoch 23 whereas Vgg16 loss touches in epoch 18 which is the highest. ResNet and DenseNet has the least loss compared to other two. In Figure 25, it is visible that ResNet and DenseNet has the least validation curve. ResNet has the least loss of 0.019 percent in epoch 17, whereas InceptionNet performs the worst at epoch 16 of 0.05 percent.

In Figure 26, it is visible that our model which is the DenseNet and ResNet which performs the best as you can see clearly compared to the other paper models which is the KNN, Logistic Regression and the Decision Tree. All of our models perform greater than 98 percentage and Decision Tree performs the least which is 90 percentage. In Figure 27, we compared all the loss value with other base paper models and it is visible that our models the DenseNet and ResNet has the least loss percentage which is close to 0.03 and whereas the Decision Tree KNN has the maximum loss percentage.

In Figure 28, we trained our model DenseNet and an older model Decision Tree with our validation Dataset with 100 epochs. We can clearly see that after 25 epochs the training accuracy starts to peak down, this is because of the Overfitting problem. When a neural network model is trained using more epochs than necessary, the training model largely learns patterns that are unique to the sample data. This prevents the model from functioning successfully on a new dataset. This model performs well on the training set (sample data), however, it performs poorly on the test set. In other words, by overfitting the training data, the model loses its ability to generalize. The model should be trained for the ideal number of epochs to reduce overfitting and improve the neural network's generalization ability. In Figure 29, we can see that loss gradually increases after 25 epochs making it lose its accuracy. To train 100 epochs, it took almost 43 hours to train a model which is quite long and not efficient. So, we

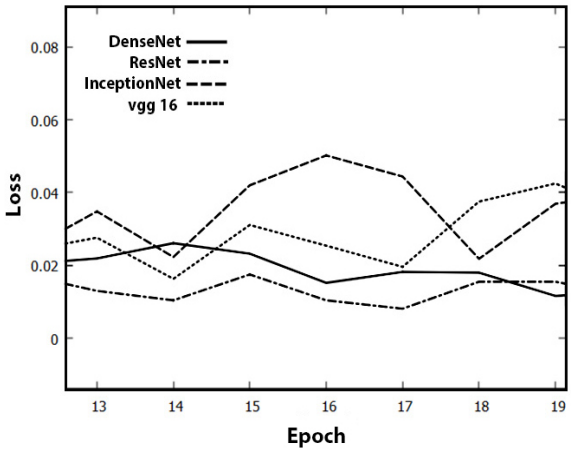

FIGURE 25: Comparison of Validation loss with the four of our Models for Dataset B

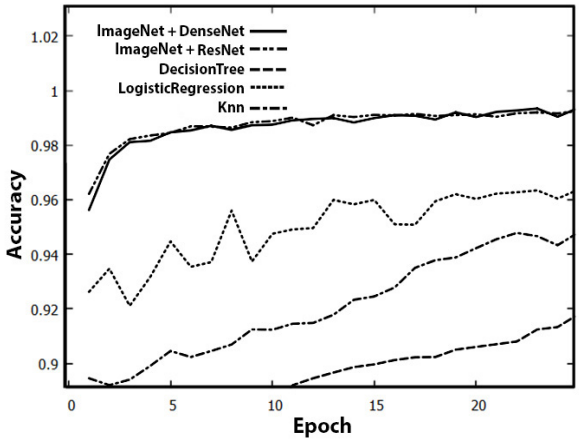

FIGURE 26: Comparing Our Models Accuracy with other Machine Learning Models for Dataset B

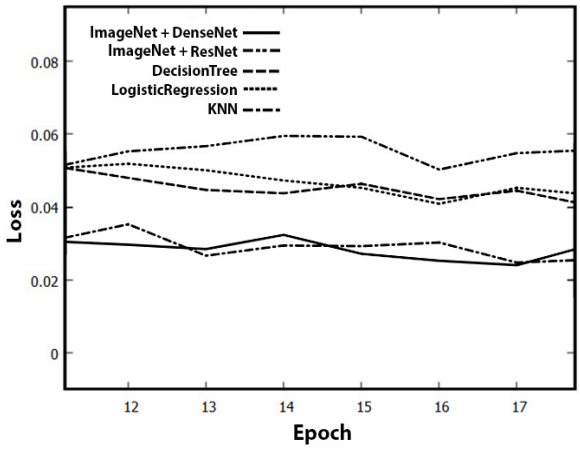

FIGURE 27: Comparing Our Models Loss with other Machine Learning Models for Dataset B

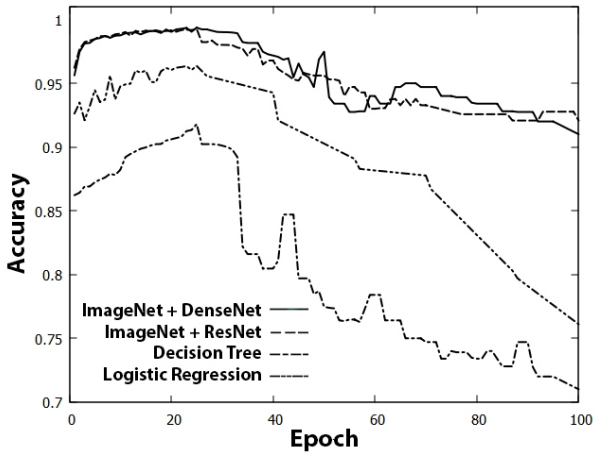

FIGURE 28: Accuracy of our Model with 100 Epochs for Dataset B

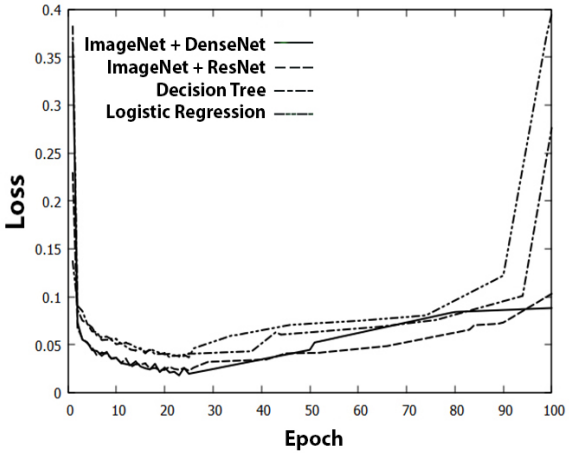

FIGURE 29: Loss Performace of our Model with 100 Epochs for Dataset B

are using 25 epochs as it is the ideal number for our dataset.

3) Testing Evaluation For Dataset C

In Figure 30, we trained our model DenseNet, ResNet, and an older model Decision Tree with our validation Dataset with 100 epochs. We can see that all the models peak up to 25 epochs and we can see the dip in the graphs after 25 epochs. DenseNet performs better in this dataset C as we can clearly see that there is less deviation compared to ResNet.

In Figure 31, we show the loss performance of our model with 100 epochs for Dataset C. We can clearly see that our

TABLE 4: Accuracy and Loss on Test Dataset B

Models	Testing Accuracy	Testing Loss
DenseNet201	98.7	4.6
InceptionV3	98.14	9.1
ResNet50	97.6	8.9
VGG16	96.8	10.9

models like DenseNet and ResNet perform better and has the least loss up to 25 epochs. Then after 25 epochs, we can see the 'U' curve that starts to affect because of the overfitting problem. So we can say that 25 epochs are the ideal number to train for our dataset.

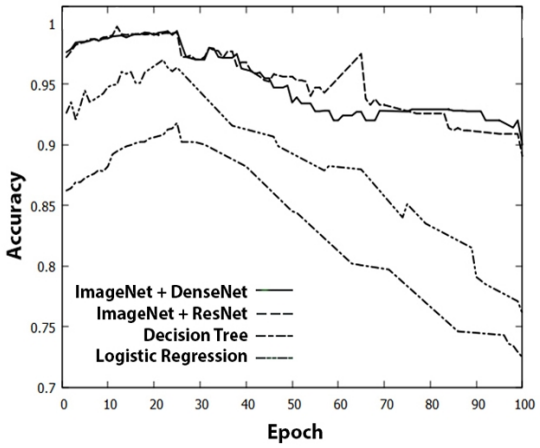

FIGURE 30: Accuracy of our Model with 100 Epochs for Dataset C

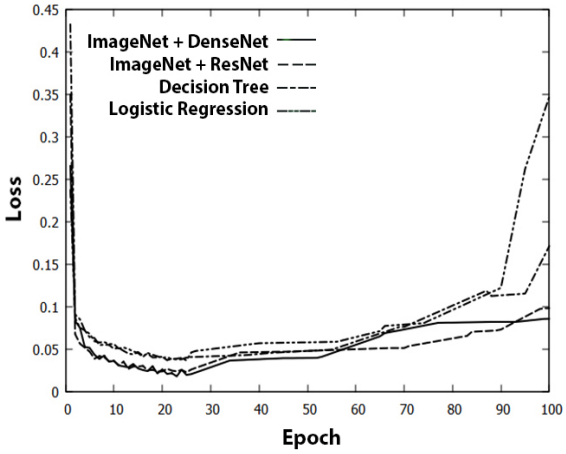

FIGURE 31: Loss Performace of our Model with 100 Epochs for Dataset C

TABLE 5: Accuracy and Loss on Test Dataset C

Models	Testing Accuracy	Testing Loss
DenseNet201	98.6	4.8
InceptionV3	97.6	8.8
ResNet50	98.2	9.8
VGG16	97.8	10.7

VI. ANALYSIS AND DISCUSSION

The Testing Dataset A has 539 photos in total, of which 256 are in the Covid class and 283 are in the normal class. The VGG16 model, which has an accuracy of 96.846% and a test loss of 10.7804%, was found to have the lowest accuracy when our models were assessed using unseen data in Table 3. As compared to the other models we considered, the DenseNet201 helped us achieve an accuracy of 98.515% and a loss of 4.894%. Classification accuracy for Inception V3 and ResNet50 is 98.1447% and 97.7737%, respectively, with test loss of 8.8486% and 9.2988%.

The testing Dataset B and Dataset C contains less images compared to Dataset A in which DenseNet Model has the highest percentage accuracy of 98.7 and 98.6 percentage comparetively as shown in Table 4 and Table 5. Whereas, VGG16 models performs the least in Dataset B and InceptionV3 performs the least in Dataset C.

ResNet Model Training accuracy and Training loss is better than DenseNet but still testing accuracy is lesser. This is because we have an overfit model. In essence, our resnet model has learned particulars that help it perform better in our training data that are not applicable to the larger data population and therefore result in worse performance.

Since Loss is more in Inception Classifier than in Optimized DenseNet Classifier, When we test it with new dataset, Optimized DenseNet Classifier works better than the Inception Classifier. The Inception Classifier Overfits the data making it lose its accuracy. Whereas Optimized DenseNet Classifier has the best fit and gives the best result.

In this experiment, we used four Deep Learning architectures—VGG16, DenseNet201, Inception V3, and Resnet50—to classify chest X-rays into Covid and normal groups. Among the four architectures mentioned above, we wanted to know which deep learning architectures performed the best in this project. We also wanted to understand their architecture and assess their performance using a variety of performance indicators, which are covered in section VI.Based on the results of our experiment, we could draw the conclusion that InceptionV3 had a classification accuracy of 98.1 and DenseNet201 had a maximum classification accuracy of 98.51%. In addition, although four-fold cross-validation results for the models InceptionV3, ResNet50, and VGG16 showed good accuracy results, their losses were nearly twice as great as those of Densenet201. On unseen or test data, the loss measure of the trained models is key. According to our research, DenseNet201 is better suited for real-world instances than other evaluated models since it has lower validation and testing loss.

VII. CONCLUSION

Since X-rays are less expensive than CT scans, our main objective for working on this study was to be able to use them to increase the reliability and accessibility of the Covid-19 prediction. Although the existing results looked promising, all of their models were only trained on a tiny dataset, and they only used one fold with an 80:20 split for assessment. All the discussed Dense Convolutional Neural Network with ImageNet for feature extraction with our base CNN were trained and tested on a sizable Covid chest X-ray dataset and were trained and evaluated using 4-fold cross-validation. This increased the reliability of the CNN models.We draw

the conclusion from section VI that we were successful in making a more accurate and dependable prediction using both validation and test data. In comparison to InceptionV3, Resnet50, and VGG16, the DeneseNet201 fared well in terms of accuracy and loss, making it more trustworthy for use with real-world data. If the trained CNN model were built into a smartphone application, it would be easier for people to use and vastly increase the number of healthcare options available in dire situations. One method to further research and enhance the model's performance is to train the model with larger and more diverse datasets. This would enable us to determine what additional elements would be necessary to increase the reliability of this model.

REFERENCES

[1] K. Roosa, Y. Lee, R. Luo, A. Kirpich, R. Rothenberg, J. M. Hyman, P. Yan, and G. Chowell, "Real-time forecasts of the COVID-19 epidemic in china from February 5th to February 24th, 2020," Infect. Dis. Model., vol. 5, pp. 256–263, Feb. 2020.

[2] L. Yan, H. T. Zhang, Y. Xiao, M. L. Wang, C. Sun, J. Liang, S. S. Li, M. Y. Zhang, Y. Q. Guo, Y. Xiao, X. C. Tang, H. S. Cao, X. Tan, N. N. Huang, B. Jiao, A. L. Luo, Z. G. Cao, H. Xu, and Y. Yuan, "Prediction of criticality in patients with severe COVID-19 infection using three clinical features: A machine learning-based prognostic model with clinical data in Wuhan," medRxiv, 2020.

[3] C Wang, PW Horby, FG Hayden and GF. Gao, "A novel corona virus out break Of global health concern"Lancet, vol. 395, pp. 470-3, 2020.

[4] Y.H.Xu,J.H.Dong,W.M.An,X.Y.Lv,X.P.Yin,J.Z.Zhang,L. Dong, X. Ma, H. J. Zhang, and B. L. Gao, "Clinical and computed tomographic imaging features of novel coronavirus pneumonia caused by SARS-CoV-2," J. Infect., vol. 80, no. 4, pp. 394–400, Apr.2020.

[5] E. Mahase, "Coronavirus: COVID-19 has killed more people than SARS and MERS combined, despite lower case fatality rate," BMJ, vol. 368, pp. m641, Feb. 2020.

[6] B. Armocida, B. Formenti, S. Ussai, F. Palestra, and E. Missoni, "The italian health system and the COVID-19 challenge," Lancet Public Health, vol. 5, no. 5, pp. e253, May 2020.

[7] A. Narin, C. Kaya, and Z. Pamuk, "Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks, " arXiv preprint arXiv: 2003.10849, 2020.

[8] Y. Li and L. M. Xia, "Coronavirus disease 2019 (COVID-19): Role of chest CT in diagnosis and management," Am. J. Roentgenol., vol. 214, no. 6, pp. 1280–1286, Jun. 2020.

[9] "Patient Page". ARRT—The American Registry of Radiologic Technologists. Available online: https://www.arrt.org/Patient-Public/Patient-Page

[10] Meng, H.; Xiong, R.; He, R.; Lin, W.; Hao, B.; Zhang, L.; Lu, Z.; Shen, X.; Fan, T.; Jiang, W.; et al. CT imaging and clinical course of asymptomatic cases with COVID-19 pneumonia at admission in Wuhan, China. J. Infect. 2020

[11] Abayomi-Alli, O.O.; Damaševičius, R.; Maskeliūnas, R.; Misra, S. An Ensemble Learning Model for COVID-19 Detection from Blood Test Samples. Sensors 2022

[12] Farooq, M.; Hafeez, A. Covid-resnet: A deep learning framework for screening of COVID-19 from radiographs. arXiv 2020

[13] Long, W.; Lu, Z.; Cui, L. Deep learning-based feature engineering for stock price movement prediction. Knowl. Based Syst. 2019

[14] X. He, X. Yang,S. Zhang, J. Zhao, Y. Zhang, E. Xing, and P Xie. "Sample-Efficient Deep Learning for COVID-19 Diagnosis Based on CT Scans," 2020

[15] S. Wang et al "A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19)," 2020

[16] M. Polsinelli, L Cinque and G. Placidi. "A Light CNN for detecting COVID 19 from CT scans of the chest ,"2022

[17] T. Anwar and S. Zakir, "Deep learning based diagnosis of COVID-19 using chest CT-scan images", 2020 IEEE 23rd International Multitopic Conference (INMIC), pp. 1-5, 2020.

[18] M. R. Islam and A. Matin, "Detection of COVID 19 from CT Image by The Novel LeNet-5 CNN Architecture", 2020 23rd International Conference on Computer and Information Technology (ICCIT), pp. 1-5, 2020

[19] I. Shiri et al., "Deep Residual Neural Network-based Standard CT Estimation from Ultra-Low Dose CT Imaging for COVID-19 Patients", 2020 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), pp. 1-3, 2020

[20] Y. Saad, A. Mustapha and A. Cherry, "Automatic classification between COVID-19 pneumonia lung cancer and normal lung tissues on chest CT Scans", 2021 Sixth International Conference on Advances in Biomedical Engineering (ICABME), pp. 197-201, 2021

[21] F. Bougourzi, R. Contino, C. Distante and A. Taleb-Ahmed, "CNRIEMN: A Deep Learning Based Approach to recognize Covid-19 from CT-Scan", ICASSP 2021 - 2021 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 8568-8572, 2021.

[22] S. Chaudhary, S. Sadbhawna, V. Jakhetiya, B. N. Subudhi, U. Baid and S. C. Guntuku, "Detecting Covid-19 and Community Acquired Pneumonia Using Chest CT Scan Images With Deep Learning", ICASSP 2021 - 2021 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 8583-8587, 2021.

[23] S. Xue and C. Abhayaratne, "Covid-19 Diagnostic Using 3d Deep Transfer Learning for Classification of Volumetric Computerised Tomography Chest Scans", ICASSP 2021 - 2021 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 8573-8577, 2022.

[24] X. Li, C. Li and D. Zhu, "COVID-MobileXpert: On-Device COVID19 Patient Triage and Follow-up using Chest X-rays", 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 1063-1067, 2020.

[25] M. Frid-Adar, R. Amer, O. Gozes, J. Nassar and H. Greenspan, "COVID-19 in CXR: From Detection and Severity Scoring to Patient Disease Monitoring", IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 6, pp. 1892-1903, June 2021.

[26] M. Fradi and M. Machhout, "Real-Time Application for Covid-19 Class Detection based CNN Architecture", 2021 IEEE International Conference on Design of Integrated Micro Nano-Systems (DTS), pp. 1-6, 2021.

[27] B.K. dis, M. Umri, Wafa Akhyari and K. Kusrini, "Detection of Covid-19 in Chest X-ray Image using CLAHE and Convolutional Neural Network", 2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS), pp. 1-5, 2020.

[28] S. Wang et al "A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19)," 2020.

[29]] X. He, X. Yang,S. Zhang, J. Zhao, Y. Zhang, E. Xing, and P Xie. "Sample-Efficient Deep Learning for COVID-19 Diagnosis Based on CT Scans," 2020

[30] Narin, A., Kaya, G., Pamuk, Z. (2020). Automatic detection of Coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks.2020.

[31] Md Mamunur Rahaman, Chen Li, Yudong Yao, Frank Kulwa, Mohammad Asadur Rahman, Qian Wang, Shouliang Qi, Fanjie Kong, Xuemin Zhu, and Xin Zhao. Identification of covid-19 samples from chest x-ray images using deep learning: A comparison of transfer learning approaches. Journal of X-ray Science and Technology, 28(5):821–839, 2020.

[32] Asmaa Abbas, Mohammed M Abdelsamea, and Mohamed Medhat Gaber. Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network. Applied Intelligence, 51(2):854–864, 2021.

[33] Maftouni, M., Law, A.C, Shen, B., Zhou, Y., Yazdi, N., and Kong, Z.J. "A Robust Ensemble-Deep Learning Model for COVID-19 Diagnosis based on an Integrated CT Scan Images Database," Proceedings of the 2021 Industrial and Systems Engineering Conference, Virtual Conference, May 22-25, 2021.

[34] "COVID-19." 2020. [Online] "http://medicalsegmentation.com/covid19/" 2021

[35] J. P. Cohen, P. Morrison, and L. Dao, "Covid-19 image data collection,"arXiv preprint arXiv: 2003.11597, 2020.

[36] COVID-19 Xrays. [Online]. Available: https://www.kaggle.com/andrewmvd/convid19-x-rays

[37] Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM. ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. IEEE CVPR 2017, ChestX-ray8Hospital-ScaleChestCVPR2017paper.pdf

[38] A. Bhandary, G. A. Prabhu, V. Rajinikanth, K. P. Thanaraj, S. C. Satapathy, D. E. Robbins, C. Shasky, Y.-D. Zhang, J. M. Tavares, and N. S. Raja, "Deep-learning framework to detect lung abnormality – a study with chest

X-ray and lung CT scan images," Pattern Recognition Letters, vol. 129, pp. 271–278, 2020.
[39] D. Yang, C. Martinez, L. Visuña, H. Khandhar, C. Bhatt, and J. Carretero, "Detection and analysis of COVID-19 in medical images using Deep Learning Techniques," Scientific Reports, vol. 11, no. 1, 2021.
[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[41] Li L., Yang Y., Liang H., Wu B. Transfer learning for establishment of recognition of COVID-19 on CT imaging using small-sized training datasets. Knowledge-Based Systems. 2021
[42] Khalid El Asnaoui Youness Chawki (2021) Using X-ray images and deep learning for automated detection of coronavirus disease, Journal of Biomolecular Structure and Dynamics

HOSAM EL-OCLA (Senior Member, IEEE) received the M.Sc. degree from the Department of Electrical Engineering, Cairo University, in1996, and the Ph.D. degree from Kyushu University, in 2001. He joined the Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan, in 1997, as a Research Student. He joined Lakehead University, in 2001, as an Assistant Professor, and has been an Associate Professor, since 2007. He has more than 90 publications in journals and international conferences.

KARTHIK AKSHAY KAMALESH (Member, IEEE) received the B.E degree in computer science engineering from the St.Joseph's College Of Engineering, India in 2020. He is currently pursuing a Master in Computer Science degree at Lakehead University Canada His research interests include Machine Learning, Deep Learning, Pattern Recognition and Image Processing.

RAJEEV SURESH (Member, IEEE)Rajeev Suresh received a B.Tech Computer Science degree from SRM Institute of Technology, India in 2020. He is currently pursuing a Masters in Computer Science at Lakehead University Canada. He has published 3 papers related to IoT and Cloud Computing in international conferences and journals. His keen research interests include Machine Learning, Deep Learning, Artificial Intelligence, IoT and Cloud Computing.