BenchCouncil Transactions on Benchmarks, Standards and Evaluations 3 (2023) 100119

Contents lists available at ScienceDirect

BenchCouncil Transactions on Benchmarks, Standards and Evaluations

journal homepage: www.keaipublishing.com/en/journals/benchcouncil-transactions-on-benchmarks-standards-and-evaluations/

Research article

CoviDetector: A transfer learning-based semi supervised approach to detect Covid-19 using CXR images

Deepraj Chowdhury [a], Anik Das [b], Ajoy Dey [c], Soham Banerjee [a], Muhammed Golec [d,e], Dimitrios Kollias [d], Mohit Kumar [f], Guneet Kaur [f], Rupinder Kaur [g], Rajesh Chand Arya [h], Gurleen Wander [i], Praneet Wander [j], Gurpreet Singh Wander [k], Ajith Kumar Parlikad [l], Sukhpal Singh Gill [d,*], Steve Uhlig [d]

[a] Department of Electronics and Communication Engineering, International Institute of Information Technology, Naya Raipur, India
[b] Department of Computer Science Engineering, RCC Institute of Information Technology, Kolkata, India
[c] Department of Electronics and Telecommunication Engineering, Jadavpur University, Jadavpur, India
[d] School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK
[e] Abdullah Gül University, Kayseri, Turkey
[f] Department of Information Technology, National Institute of Technology, Jalandhar, Punjab, India
[g] Department of Science, Kings Education, London, UK
[h] Department of Cardiac Anaesthesia, Hero DMC Heart Institute, unit Dayanand Medical College and Hospital, Ludhiana, Punjab, India
[i] Chelsea and Westminster Hospital, NHS Trust London, London, UK
[j] St Mary's Hospital, Trinity Health of New England, Waterbury, CT, USA
[k] Department of Cardiology, Hero DMC Heart Institute, Dayanand Medical College and Hospital, Ludhiana, Punjab, India
[l] Institute for Manufacturing, Department of Engineering, University of Cambridge, Cambridge, UK

ARTICLE INFO

Keywords:
Machine learning
Deep neural network
Transfer learning
Android app
Chest X-ray (CXR)
COVID-19
Healthcare

ABSTRACT

COVID-19 was one of the deadliest and most infectious illnesses of this century. Research has been done to decrease pandemic deaths and slow down its spread. COVID-19 detection investigations have utilised Chest X-ray (CXR) images with deep learning techniques with its sensitivity in identifying pneumonic alterations. However, CXR images are not publicly available due to users' privacy concerns, resulting in a challenge to train a highly accurate deep learning model from scratch. Therefore, we proposed **CoviDetector**, a new semi-supervised approach based on transfer learning and clustering, which displays improved performance and requires less training data. CXR images are given as input to this model, and individuals are categorised into three classes: (1) COVID-19 positive; (2) Viral pneumonia; and (3) Normal. The performance of CoviDetector has been evaluated on four different datasets, achieving over 99% accuracy on them. Additionally, we generate heatmaps utilising Grad-CAM and overlay them on the CXR images to present the highlighted areas that were deciding factors in detecting COVID-19. Finally, we developed an Android app to offer a user-friendly interface. We release the code, datasets and results' scripts of CoviDetector for reproducibility purposes; they are available at: https://github.com/dasanik2001/CoviDetector

1. Introduction

The COVID-19 virus, the first case of which is thought to have emerged in December 2019, has caused 6.9M deaths as of July 02, 2023 [1]. When infected people are coughing or sneezing, viral droplets can stay in the air for three hours. Respiratory infection harms healthy people's lungs and tissues [2]. COVID-19 variant instances have increased rapidly in recent months [3]. However, some international researchers think that this increase will decrease by 2024, and the world may become normal [4]. At present, COVID-19 detection is being performed using one of the below three tests:

* Correspondence to: School of Electronic Engineering and Computer Science, Queen Mary University of London, London, E1 4NS, UK.
E-mail addresses: deepraj19101@iiitnr.edu.in (D. Chowdhury), anikdas.2113@gmail.com (A. Das), deyajoy80@gmail.com (A. Dey), soham20101@iiitnr.edu.in (S. Banerjee), m.golec@qmul.ac.uk (M. Golec), d.kollias@qmul.ac.uk (D. Kollias), kumarmohit@nitj.ac.in (M. Kumar), guneetkw.it.22@nitj.ac.in (G. Kaur), rupinderchem@gmail.com (R. Kaur), drrajesharya@yahoo.com (R.C. Arya), gurleenwander@gmail.com (G. Wander), praneet_wander@hotmail.com (P. Wander), drgswander@yahoo.com (G.S. Wander), aknp2@cam.ac.uk (A.K. Parlikad), s.s.gill@qmul.ac.uk (S.S. Gill), steve.uhlig@qmul.ac.uk (S. Uhlig).

https://doi.org/10.1016/j.tbench.2023.100119
Received 6 June 2023; Received in revised form 3 July 2023; Accepted 3 July 2023
Available online 6 July 2023

- Computed Tomography (CT) scans of chest that use three-dimensional radiographs and are a key diagnostic tool. However, not every health care institution has the facility of CT scan with them.
- Ribonucleic Acid (RNA), which can be detected from nasopharyngeal swabs using the Reverse Transcription Polymerase Chain Reaction (RT-PCR) technique. However, this technology is hard to get to, and is time consuming.
- Chest X-ray (CXR); the necessary equipment for a CXR is readily available and is more transferable and quick compared to CT-scan ones. CXR examinations are also fast as they require only roughly 15 s for each participant.

As COVID-19 continues to sweep the globe, researchers and data scientists have begun employing deep learning methods for the automated identification of the virus in humans [5]. Artificial Intelligence (AI)-based computer-aided diagnostics has advanced rapidly in several domains of medicine because of contemporary advances in the field of AI [6]. Diseases like cancer can be diagnosed with greater accuracy thanks to automatic image analysis performed via deep learning and in more detail Convolutional Neural Networks (CNN) [2]. Therefore, these models show promise for enhancing the use of CXR images in the diagnosis of COVID-19 [7].

AI systems have previously been utilised to correctly identify pneumonia from CXR [8]. Differentiating between viral and bacterial pneumonia has been performed via the use of deep learning. K-Nearest Neighbour (KNN), Support Vector Machine (SVM), and CNNs are only a few of the AI classification strategies used [9]. Among the machine learning algorithms for classification tasks, KNN is considered one of the easiest [10]. The KNN algorithm basically considers the similarity distance between the new and already available data and classifies the new data accordingly [11]. SVM is another classification algorithm, that primarily creates the best possible boundary that can separate n-dimensional space into classes with the aim to classify the new data into one of the classes [12]. CNN is effective in image classification to recognise COVID-19 [13–15]. Multi-layer neural networks, like CNNs, are the key to the system's success in recognising visual trends. Various pre-trained CNN models, including AlexNet, VGG16, InceptionV3, and DenseNet are available, among which InceptionV3 demonstrates better accuracy and performance for the COVID-19 classification [16].

1.1. Motivation

CXR-based identification of COVID-19 patients might be hampered by a lack of effective and experienced medical experts, especially in rural areas [17]. So, there is a requirement for a simple and inexpensive deep learning-based technique to identify COVID-19 patients within a short span of time [18]. This model will be available to all patients, even though doctors may not be available [19].

CXR of COVID-19-affected lungs show less porosity or visibility because the lungs are stuffed with smooth and dense mucus [20]. While multiple algorithms and diagnostic tools based on machine and deep learning have shown promise, they still fall short of high performance in terms of precision and error rate [21]. Therefore, healthcare professionals and the community as a whole would benefit from choosing a group of effective deep learning-based analysts.

In this paper, CoviDetector is a machine learning system that utilises transfer learning and a deep learning model (InceptionV3) for the early identification and diagnosis of COVID-19 by chest X-rays. Moreover, this system is deployed as an Android Backend. The android application user interface (UI) is designed using the React Native framework, which takes as input of an image and predicts and displays the results on the screen.

CoviDetector aims to provide an Android application for the user, which will allow the user to predict COVID-19 automatically using CXR images. In order to have quick, accurate, economical, and hassle-free COVID-19 recognition, we compare the performance of various deep learning models across various metrics, including accuracy, precision, recall, F-score, and sensitivity, with the ultimate goal of applying the most effective model on the back-end of an Android app.

1.2. Problem statement

Detection of COVID-19 can be done with a RT-PCR test, which can only be conducted in a laboratory environment; tests are performed by taking a swab from the nose. The time required for this test's results varies between 8 and 24 h. Therefore, alternative methods, such as rapid antigen testing, have emerged. However rapid antigen tests are not accepted by many medical practitioners because of their high false positive and false negative rates. For this reason, a COVID-19 detection method that takes as input CXR images is used; this method is both fast and reliable. In this paper, we propose a smartphone application that detects COVID-19 using deep learning (DL) and CXR images (as an alternative to RT-PCR tests). Users can quickly learn if they have COVID-19 by uploading CXR images to the Android app. We apply several DL algorithms to CXR images to detect COVID-19. These models are trained with unbiased and balanced data, so the prediction results are unbiased and accurate. Experimental results have shown that the models display high performance in detecting COVID-19.

1.3. Our contributions

The main contributions of this work are:

- an Android application compatible with smartphones that diagnoses COVID-19 through CXR images. The diagnosis is performed from CXRs via the proposed semi-supervised method that consists of a Deep Neural Network (DNN) and K-means clustering; the proposed methodology also utilises transfer learning. Finally, GradCAM is used to highlight the areas in the CXRs that were the deciding factors in the method's decision in detecting COVID-19;
- the DNN model trained with InceptionV3 CNN blocks is the best performing model for detecting COVID-19.

CoviDetector is an Android app in development with the intention of providing medical professionals and consumers with an easy way to check for Covid-19. The proposed App helps patients receive proper classification results for the Chest X-rays uploaded by them. This would primarily help the doctors take immediate action without waiting for reports and start instant treatment. CoviDetector can accurately determine if a person is COVID-19 positive or negative by analysing only an image of CXR. This information, as we have previously mentioned, can be used by doctors to make instant decisions. This would also help society, as the resources utilised are just an Internet connection with no manpower or industrial interference.

The rest of the paper is organised as follows: Section 2 presents related work. Section 3 introduces a proposed methodology, and Section 4 presents the experimental setup and results. Finally, Section 5 concludes the paper and highlights future directions.

2. Related work

Over the past several months, a growing body of research has evaluated the efficacy of deep-learning models for the identification of COVID-19 in CXR images. This section includes short discussions of a few of these works that are relevant to our own.

Accuracy of 98.93%, specificity of 98.66%, precision of 96.39%, and F-1 score of 98.15% were achieved with the approach described by Mittal et al. [22]. There were a total of 1215 images in their collection, including 250 from COVID-19. COVIDiagnosis-Net was developed by Ucar et al. [23], and the precision across all three classes was 98.26%. The DNN model presented by Ahammed et al. [24] has achieved 94.03 percent accuracy using the CNN. The researchers have used data from

Table 1
Comparative study of relate works for COVID-19 detection.

Works	Dataset used	Methodology	Accuracy(%)	Android app	Transfer learning	Training data Size
Xinggang et al. [28]	Custom Gathered Dataset	UNet based 3D DNN	90.10%	✗	✓	229 No-Findings & 313 Covid-19
Yujin et al. [29]	CoronaHack & Other Datasets	ResNet18 based FC-DenseNet	91.90%	✗	✓	134 No-Findings & 126 Covid-19 & 94 Others
Ahmed et al. [30]	IEEE CovidCXR Dataset	CNN based Approach	94.00%	✓	✗	1341 No-Findings & 1200 Covid-19
Ozturk et al. [26]	CohenJP Dataset	DarkNet-19 based CNN	98.08%	✗	✗	500 No-Findings & 125 Covid-19 & 500 Pneumonic
Ucar et al. [23]	CovidX Dataset	Deep Bayes-SqueezeNet based Approach	98.26%	✗	✗	1229 No-Findings & 1229 Covid-19 & 1229 Others
Bushra et al. [31]	CohenJP & other datasets	Tensorflow Lite based CNN	98.65%	✓	✗	592 No-Findings & 592 Covid-19
Taresh et al. [32]	COVID-19 Radiography Database	VGG16 based CNN	98.72%	✗	✓	1140 No-Findings & 820 Covid-19 & 1150 Pneumonic
Ahsan et al. [33]	CohenJP & CovidCXR Datasets	Feature Fusion based CNN	99.49%	✗	✗	2489 No-Findings & 1584 Covid-19
CoviDetector (This Paper)	**CovidCXR, NIH CXR, DLAI3 datasets**	**InceptionV3 based CNN and Clustering**	**99.69%**	✓	✓	4253 No-Findings & 3160 Covid-19 & 6034 Others

three categories to train the algorithm. In this case, too, the dataset had a rather low sample size, which is not optimal for trying to train a deep learning-based system for COVID-19 diagnosis.

The ResNet101 CNN model was also employed by Azemin et al. [25]. The result of their work was a thousand images that were fed into the pre-trained model. They were just 71.9% accurate at best. Ozturk et al. [26] combined DarkCovidNet with a collection of 1125 photos, 125 of which were taken from COVID-19 examples, to develop a framework. An overall accuracy of 98.08 percent was found in a 5-fold cross-validation of binary tags. Utilising algorithms like ResNet50, VGG16, VGG19, and DensNet121, Khan et al. [27] constructed a novel framework for diagnosis of CXR images; VGG16 and VGG19 demonstrated higher accuracies of approximately 99.3 percent in contrast to others.

Yujin et al. [29] employed a patch-based CNN technique for a much lesser amount of trainable parameters for COVID-19 diagnosis, which they attributed to their use of a segmented network-based approach. When taking into account the increased sensitivity of their line of work, the 91.9% correctness they attained is rather impressive. In a related paper, Fan et al. [34] developed Inf-Net with a parallel partial decoder to combine characteristics at a higher level. Their method was 97.4 percent accurate while also being 87.1 percent sensitive. To forecast COVID-19 CT scans, Xinggang et al. [28] also presented a 3D deep neural network. The precision of their work was 90.1%. One CT volume from a single patient was processed by the algorithm in just 1.93 s, making it one of the quickest models ever created.

A further investigation aimed to identify COVID-19 using a transfer learning strategy and three pertained models was conducted by Loey et al. [35]. Correctness for all three categories on AlexNet was 85.20 percent using a dataset of over 300 X-ray pictures that included around 70 photos of COVID-19. In order to enhance the ResNet-101 and ResNet-151's weights during training, fusion effects were used, and Wang et al. [36] were able to increase the model's accuracy to 96.1%. Mahmud et al. [37] also obtained a success rate of 97.4 percent for binary classes using a CNN model they devised called CovXNet. A deep learning method was described by Minaee et al. [38] to identify COVID-19 from CXR. Using data augmentation, they generated modified pictures of the CXR plates, and their approach had a sensitivity of 98% and a specificity of 90%.

In another study on COVID-19 detection, Chakrabarti et al. [32] employed ensemble learning in conjunction with a Deep Convolutional Neural Networks (DCNN) to predict binary classes. They employed an aggregate of 1006 COVID-19 suspected patient's pictures (538 positive and 468 negative) to evaluate the performance of the model. The degree of precision they achieved was 91.62 percent. Ahmed et al. [30] also used Tflite to develop a method for developing mobile applications that relied on CNNs. They found that their method, on average, was 94% accurate. Ahsan et al. [33] found similar success with feature fusion and deep learning. The recommended approach improved accuracy to 99.49 percent, performing better than any single CNN.

To boost the overall performance of COVID-19 methods of classification, Tabik et al. [39] used a Smart Data Based network called COVID-SDNet. Their method had a 97.72 percent success rate. Taresh et al. [40] utilised transfer learning to identify COVID-19 from CXR images, so it is possible to do the same. With a 98.72 percent accuracy rate, their VGG16-based model was superior to the competition.

2.1. Critical analysis

Table 1 compares **CoviDetector** with existing transfer learning and DNN models. The accuracy of the aforementioned study works is pretty high, yet it is also important to put it into practise in a way that could be accessible to patients in general. Only two of the reported studies have even investigated using transfer learning in a user interface for an application supported by the model. A novel framework that enhances accuracy and implementation in an Android App is required as very little work has been done to construct a CNN-based Android App, specifically for COVID-19 diagnosis. The required framework needs to be proposed in order to (1) facilitate communication between users, (2) incorporate the most effective method with better precision, and (3) guarantee the highest level of care for both patients and medical professionals. To fill these knowledge shortcomings, we propose a novel system called CoviDetector to improve research and address the above-mentioned challenges.

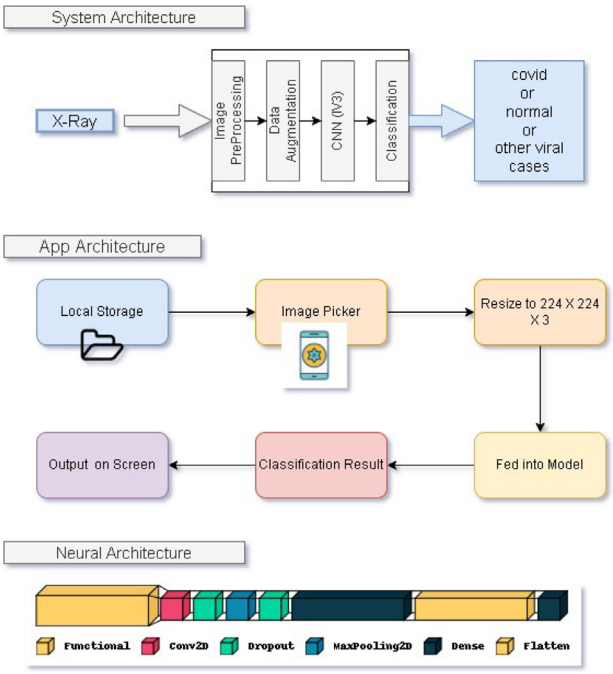

Fig. 1. CoviDetector Architecture: The first portion of the figure depicts the layers in the System Architecture i.e the basic internal working of the CoviDetector application. The next portion is the App Architecture which shows the functioning of the Application UI to fetch the inputs from the user and how it is fed to the model for evaluation. The next and last portion of the image picturises the sequence of layers in the Neural Architecture used in the CoviDetector model.

3. CoviDetector: Proposed methodology

Methods for accomplishing the goals of the proposed work are discussed in detail below, with specific attention paid to the system architecture that was developed to accomplish these goals including input preprocessing of the image inputs followed by labelling appropriate classes of data, and datasets used, continuing with the extraction of features, to CNN-based classification.

3.1. System architecture

CXR images are used as input for the proposed technique for COVID-19 detection. To begin, this system shifts user-provided photos into the more widely used Red–Green–Blue (RGB) colour space. Additionally, the algorithm only takes into account photos that are comparable to CXR. To begin, a Structural Similarity Index Measure (SSIM) is applied to the picture in order to determine its structural similarity with a CXR. If that happens, just that picture will be considered in further analyses. For example, if the image has a SSIM value greater than the threshold value (this value depends on the type of application) with the image of CXR, then that image will be considered for prediction; otherwise, that image will be discarded. The InceptionV3 model is quite effective in obtaining features and picture classification. The whole model was implemented using an Android Front-end made using the React-Native framework and TensorflowJS as a backend. The System Architecture has been visualised in Fig. 1. The system takes CXR images as input, classifies them into different classes, and gives the predicted class as the output.

3.2. Input pre-processing

Image pre-processing is a vital step to achieving meaningful and accurate classification. Therefore, there is a need to resize all images to $224 \times 224 \times 3$ pixel resolution and their intensity values to be normalised to $[-1, 1]$ (by dividing with 255). Subsequently an 80%–20% ratio between training and testing sets have been applied to every dataset.

3.3. Data augmentation

The dataset included a highly imbalanced number of samples from various classes. We first used the data augmentation technique to increase the number of sample images for every class in order to address the class imbalance. Random cropping and horizontal & vertical flips were applied for the augmentation of existing data. In order to equalise the number of samples from each class, the following stage included selecting the class with the fewest images and extracting random samples from other classes. The researchers were able to create a more optimal model using the larger sample size. The size of the training dataset has been increased by data augmentation. Further, random cropping, and horizontal & vertical flips have been applied to improve the robustness of the training model.

3.4. Model

This section gives an overview about deep learning models used in this research work.

3.4.1. VGG16

The VGG16 network is a CNN model with 1000 outcomes. It has 13 convolution neural layers and three layers that are fully connected. It is capable of handling 224×224 pixel colour pictures. After that, many convolution networks are used to determine if the layer is red, green, or blue. Both the input and the resultant feature maps have the same size in this scenario. The field of reception of any convolution filter is just 3×3 in stride 1. Row and column padding are used following convolution to preserve spatial resolution. There are 13 convolution layers and 5 max pool layers, as has already been stated. The largest pooling window is 2 strides by 2 strides. VGG16's architecture and primary feature, transfer learning, are both formed by [41]. For the purpose of using CNNs as a fixed feature extractor1, a CNN architecture trained on a large dataset is taken, and its final fully connected layer is removed. For this new dataset, the remainder of the CNN serves as a fixed-feature extractor. Let us assume that there is a model that is trained on the basics of one set of databases, like ImageNet and an application that is trained on the basics of some other database, like Pascal. When an image enters the first layer, consider only the edge. Then it moves to the second layer, which considers corners, curves, etc. On further moving to the third layer, it considers the features of the high-level layer. On further moving more deeper, the domain becomes more specific.

3.4.2. VGG19

To put it simply, VGG19 is a state-of-the-art convolutional neural network. The visual geometry group at Oxford has put forth this idea. Including its 16 convolutional layers, 3 fully connected layers, 5 max pool layers, and 1 softmax layer, the VGG19 model is rather complex. This phenomenon has been designated as VGG19. It has been taught with millions of different photos, giving it a great depth of understanding. Colour photos of a certain width and height are acceptable. After that, the pictures go through a series of convolution networks, one for each colour channel. Both the input and the resultant feature maps have the same size in this situation. The receptive field size of every convolution filter is exactly 3×3 stride 1. Use row and column padding after convolution to keep spatial resolution stable. The architecture of the network consists of 13 convolution layers and 5 max pool layers, as stated before. The largest allowable pooling window size is 2 by 2 steps. VGG19 borrows its model architecture on like its predecessor, VGG16. When pitted against VGG16, VGG19 performs somewhat better. It represents an idea in terms of form, colour, and architecture. If the picture is in the ImageNet database, a pre-trained version of the network can be loaded and used. After being taught, the network can sort photos into one of a thousand distinct categories, each of which can include commonplace items like a computer keyboard, mouse, or pencil.

3.4.3. DenseNet121

The DenseNet121 model is part of the larger DenseNet family of image classifiers. The DenseNet121 model differs mostly in terms of its larger size and greater precision. The DenseNet121 model is considerably bigger than its smaller counterpart. They were originally developed using Torch, but the authors have since ported them to Caffe. Pre-training on ImageNet has been done for the DenseNet models. DenseNet121's model results are representative of those of an object classifier applied to a dataset of 1000 classes from the ImageNet database. A single picture with the coordinates (1, 3, 224) in BGR order is the input to the simulation. Fewer interconnections between layers near the input and the ones near the output allow convolutional networks to be significantly deeper, more precise, and simpler to train, as proven in recent research [42]. The article employs a feed-forward neural network architecture called a Dense Convolutional Network (DenseNet). Layer data for DenseNet121 has been retrieved from [42]. Every level feeds its own feature maps into the layers above it, and each layer above it feeds its own feature maps into the levels below it. Using DenseNets has been shown to drastically cut down on the number of parameters. DenseNets achieves great performance with less memory and compute while significantly outperforming the latest developments on the majority of them.

3.4.4. InceptionV3

The InceptionV3 transfer learning method using weights from publicly available ImageNet data will serve as the focus of this research. There are 230 "frozen" layers in the model, representing parameters that should not be modified throughout the training process. Developing a model from preexisting models was shown to be more efficient than developing a new deep learning model from beginning [43]. Transfer learning allows us to retrain the final layer of an existing model, resulting in a significant decrease in not only training time, but also the size of the dataset required. One of the most known models that can be used for transfer learning is Inception V3. This model was originally trained on over a million images from 1000 classes on some very powerful machines, which resulted in highly accurate classification. Inception V3 mainly centres on consuming less computational power by modifying the previous Inception architectures. Compared to VGGNet, Inception Networks (GoogLeNet/Inception v1) have proven to be more computationally practical, both in terms of the number of parameters generated by the network and the memory and other resources used.

3.4.5. EfficientNet

EfficientNet, which was first introduced in Tan and Le's 2019 paper [44], is one of the best algorithms for typical image categorisation transfer learning tasks and ImageNet, where it has achieved State-of-the-Art efficiency. In terms of model size, the lowest base model is competitive with MnasNet, which obtained near-SOTA with a much smaller model. EfficientNet introduces a heuristic approach to model scaling, producing a set of models (B0–B7) that strike a good balance between quickness and precision when the scale is increased or decreased. However, numerous factors limit the resolution, depth, and width options. Resolutions that are not divisible by eight, sixteen, or twenty-four result in zero-padding along the end points of some layers, wasting computational resources. This is particularly true for the model's smaller variations, which is why the input resolutions for B0 and B1 are set to 224 and 240, respectively. EfficientNet's construction blocks require channel sizes to be multiples of eight. When depth and width can still be increased, memory constraints may stifle resolution. In this case, increasing depth or width while maintaining resolution may still increase performance.

3.4.6. K-means clustering

Clustering is a popular interactive data analysis method for quickly understanding how the data is organised. Data clustering is the process of identifying groupings within a dataset where individual data points have many similarities but those corresponding to different clusters have few. Using iterative steps, the K-means approach seeks to divide the dataset into K distinct, non-overlapping subgroups (clusters), with each cluster containing a single value. It makes an effort to keep clusters as dissimilar (far) as practicable while maintaining intra-cluster data points as closely related as reasonable. The algorithm groups data points into clusters with the goal of minimising the sum of squared distances among them and the centroid of the cluster (the mathematical average of all data points in that group). The homogeneity (similarity) of data points within a cluster increases as inter-cluster variation decreases.

3.5. Algorithm

In this research, we use transfer learning to train a CNN Model. The weights that are shared between model layers serve as a means of information transfer. A Convolutional 2D layer with ReLu activation and a Dropout layer follow this. This brings the total number of layers to 5. A MaxPooling Layer comes next, followed by a Flatten Layer for communication. The necessary number of classes with output is then sent to a softmax-activated dense layer that serves as the ultimate output layer. The following phase was to assemble the model with two primary variables called optimiser and loss. It has become common practise to use Binary CrossEntropy as the loss function for binary classification jobs, and Categorical CrossEntropy for multiclass data. It turned out that a learning rate of 0.0001 yielded the best outcomes from the RMSprop optimiser. The algorithm for the configuration of the model is visualised in Algorithm 1.

Algorithm 1 Model Input and Architecture of CoviDetector:

Require: a : $data$, b : $labels$, z : *number of images*, m, n : *image dimensions*,
f : *Base Model*, g : *Head Model*
$f.out$: *Output Layers of Functional Model*

for i=0 to z-1 **do**
  $b \Leftarrow image$
  $image \Leftarrow image_{cvtcolor}$
  $image \Leftarrow image_{resige}(m, n)$
  $a \Leftarrow image$
  i++
end for

Model(a):
  $f \Leftarrow$ VGG16,VGG19;DenseNet121; InceptionV3; EfficientNetB4
  $g \Leftarrow$ f.out(output layer of f)
  $g \Leftarrow$ Conv2D(g)
  $g \Leftarrow$ Dropout(g)
  $g \Leftarrow$ MaxPool2D(g)
  $g \Leftarrow$ Flatten(g)
  $g \Leftarrow$ Dense(g)
  return metric

3.6. The prototype application

In line with the proposed model, an Android-based mobile app has been created to distinguish between Covid-19 positive and negative patients. Because of this, anyone may access a CXR picture on their computer and input it into the programme. The image is then assessed by the programme using the provided model, and a classification label is returned. The user interface prototype is shown in Fig. 2 .

Table 2
Comparative view of data splitting for multiple datasets:.

Dataset	Training (70%)	Testing (20%)	Validation (10%)	Used samples
COVID-19 CXRImage Dataset (Research)	1125	321	160	1608 out of 1823
DLAI3 Hackathon Phase3COVID-19 CXR Challenge	762	218	110	1089 out of 5507
COVID-19 RadiographyDatabase	7140	2040	1020	10.2k out of 42.3k
Covid19 Detection	7560	2160	1080	10.8k out of 24.8k

Table 3
Performance metrics 1 (Dataset1).

Model	MCC	Sensitivity	Specificity	AUC score	Training time (sec/epoch)
VGG16	0.9667	0.9717	1.0000	0.9858	31
VGG19	0.9668	0.9811	1.0000	0.9905	36
DenseNet121	0.9853	1.0000	0.9917	0.9948	30
InceptionV3	0.9876	1.0000	1.0000	0.9959	24
EfficientNetB4	0.7635	0.8361	0.8695	0.8571	40
Semi-supervised	0.9916	0.9958	0.9962	0.9937	20

Fig. 2. The Prototype UI.

3.7. GradCAM

GradCAM is a kind of post-hoc attention. The term post-hoc attention means it is a method used for heatmap generation that is subsequently applied to a pre-trained neural network after training is complete and weights are known. GradCAM is a generalisation of CAM (Class Activation Mapping), and it can be applied to any CNN architecture. The basic idea behind the usage of GradCAM over here is to exploit the spatial information preserved using convolutional layers, in order to comprehend which parts of the input image played a pivotal role in the classification decision. It uses a feature map produced by the last convolutional layer of a CNN architecture (like CAM). We have applied GradCAM visualisation to DenseNet21, VGG16, and InceptionV3 algorithms, which were among the top-performing models for the datasets used. A few more reasons to use GradCAM are: (1) It does not change the architecture of the model and just gets added to it, and (2) It is class-discriminative using localisation techniques.

4. Performance evaluation

We have evaluated four different DNN models on the dataset and then analysed the accuracies obtained using the discussed approach. We examined the model's efficiency using a variety of loss functions and parameter settings before settling on a good one. As a last step, we integrated the Tensorflow backend into an Android app; the details are presented next.

4.1. Experimental setup

Accuracy, sensitivity, and specificity were taken into account for analysis in order to evaluate the effectiveness of several models and obtain the most effective outcomes. After training for around 20 epochs using the RMSProp optimiser and a Learning Rate of 0.0001, all of the CNN models were ready for testing. Model training takes between 31 and 35 s per epoch on VGG16 and VGG19, respectively. In InceptionV3, the time required for each epoch was around 24 s, but in DenseNet, the time required was approximately 30 s. EfficientNetB4 took about 40 s per epoch for the same data. Table 2 gives the insight of training testing and validation ratio of the datasets used. Table 3 summarises the average training time of the models.

4.2. Configuration settings

We developed the model using Tensorflow 2.2.0 and Python 3.6. NVIDIA Tesla K80 GPU has been used for the training procedure.

4.3. Dataset used

We used various datasets for experimental purposes. The authors of the datasets have accomplished the hectic task of gathering and categorising the CXR Images. There are four datasets on which experiments were performed. Two of them contain data from three classes. The other two datasets consist of data from 4 and 5 different classes. The first dataset, COVID-19 CXR Image Dataset (Research) [45] named as Dataset1 contains classes named COVID, Normal, and Viral. This dataset was used for initial model testing and validation. However, various other datasets were utilised to check the competency of the model. Another other 3 class dataset, DLAI3 Hackathon Phase3 COVID-19 CXR Challenge [46] named Dataset2 contains COVID, No-FINDING, ThoraxDisease. This dataset was also used for the validation and testing of the DNN models. The other dataset, COVID-19 Radiography

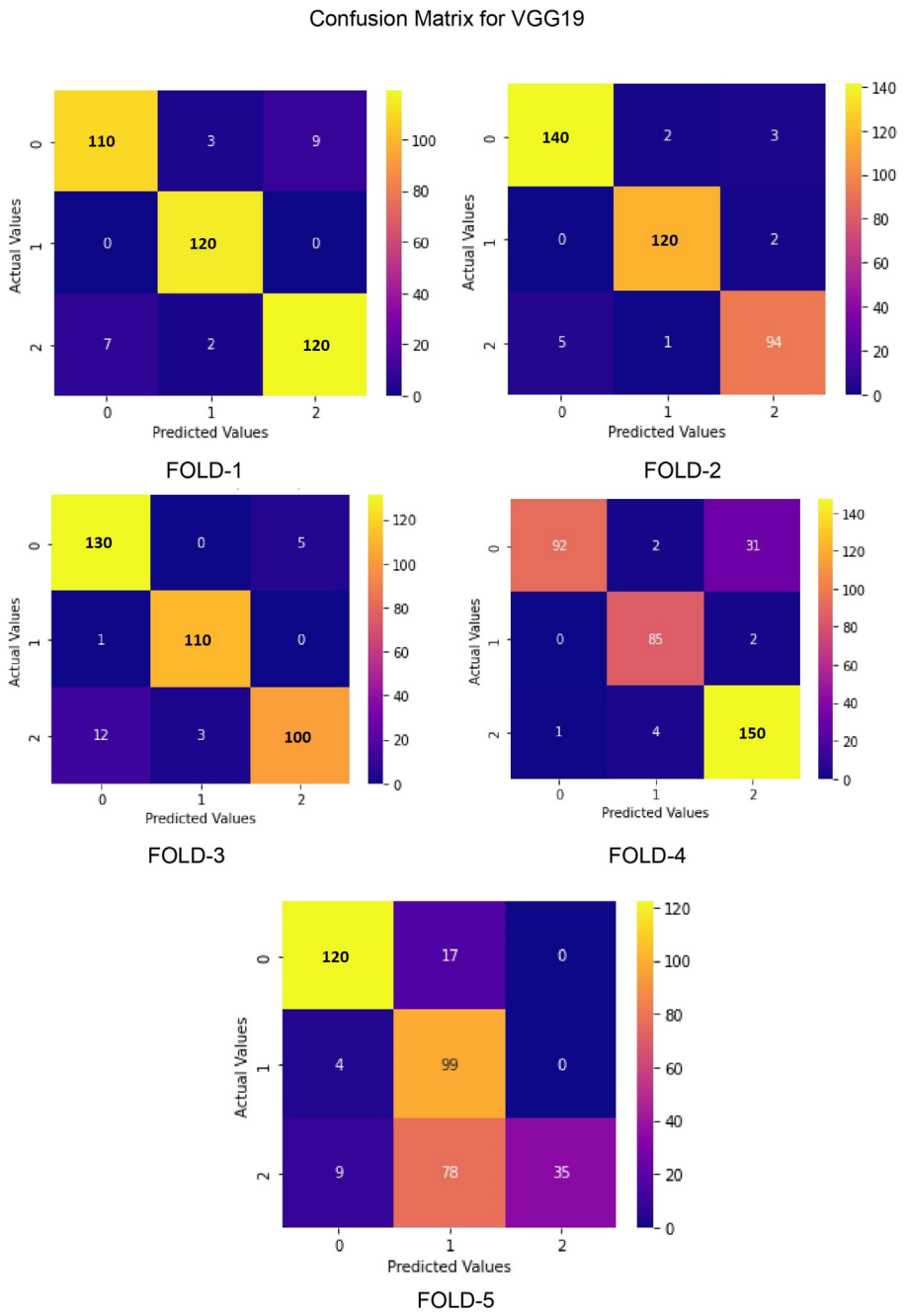

Fig. 3. Confusion Matrix:VGG19 on 5-Folds(0-Normal, 1-COVID19,2-Viral).

Database [47] named Dataset3 contains 4 classes of data namely COVID, Lung Opacity, No-Finding, and Viral Pneumonia. The best model was trained with these four dataset data points where two classes were merged to form a single class of data. This model was further evaluated on the training data and on a 5 class dataset, Covid19 Detection [48] named Dataset4 which consists of data from COVID, Fibrosis, Normal, Viral and Pneumonia, to further prove the competency of the work. Apart from that, all the datasets were split into training and testing sets with an 80:20% ratio and used for 5-fold cross-validation. The performances of four different models applied to these datasets are shown in Figs. 3–6. As can be seen in Figs. 3–7 the accuracy rate of the model decreases as the number of fold operations increases. For example, the accuracy rates for Figs. 3–5 fold-1 process are 94%, 97.29% and 95.06%, respectively. For the same shapes, these rates decrease to 91.71%, 94.02% and 90.95%, respectively, after the fold-5 process.

4.4. Analysis of results

The findings of the experiments are analysed and discussed below:

4.4.1. Results on 3 class dataset

As soon as it comes to COVID-19 detection, a True Positive (TP) happens if both the patient's other investigations and the model agree that COVID-19 is present, whereas a True Negative (TN) occurs when both the patient's other investigations and the model agree that COVID-19 is not present. If a person does not have COVID-19 but the model predicts positive, we say that person has a False Positive (FP), whereas if the person possesses COVID-19, we say that the model gets a False Negative (FN).

$$Accuracy = \frac{TP + TN}{TN + TP + FN + FP} \tag{1}$$

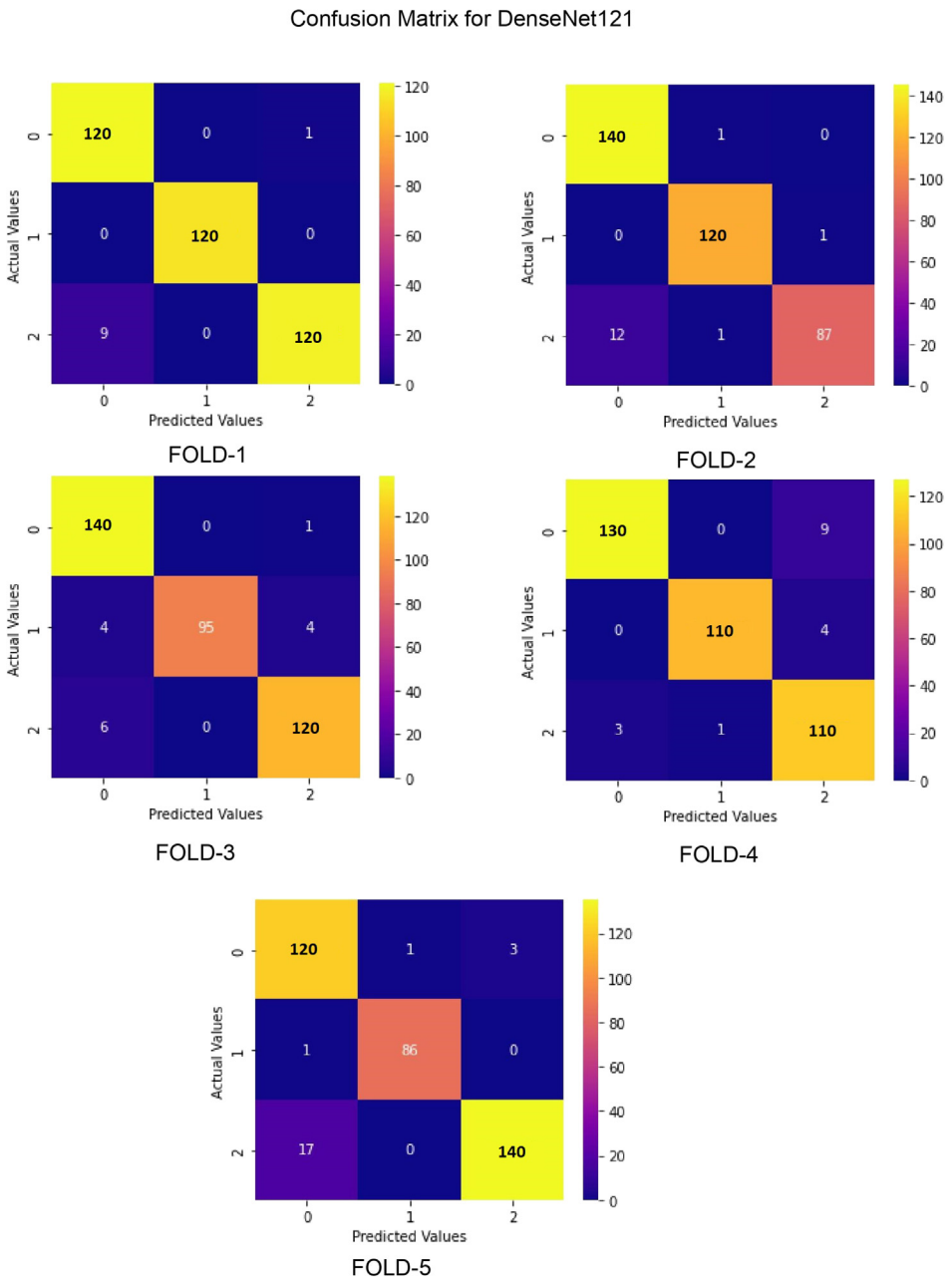

Fig. 4. Confusion Matrix:DenseNet121 on 5-Folds(0-Normal, 1-COVID19,2-Viral).

$$Sensitivity = \frac{TP}{TP + FN} \tag{2}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$F - Score = \frac{TP}{TP + (0.5)(FN + FP)} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{7}$$

The data visualisations show a consistent trend, with training and testing accuracy improving and loss decreasing as the number of epochs grows.

Table 4
Performance metrics 2 (Dataset1).

Model	Accuracy	Precision	Recall	F-Score
VGG16	0.9876	1.0000	1.0000	1.0000
VGG19	0.9917	1.0000	1.0000	1.0000
DenseNet121	0.9958	1.0000	0.9917	0.9958
InceptionV3	0.9965	1.0000	1.0000	1.0000
EfficientNetB4	0.7863	0.8026	0.7685	0.7553
Semi-supervised	0.9969	0.9989	1.000	0.9971

Table 3 shows the MCC (Matthews correlation coefficient), sensitivity, specificity, and AUC Score values for each model (see 4.4.3 for Semi-supervised model). Whereas accuracy, precision, recall and F1 score are also tabulated in Table 4.

Fig. 8 not only depicts the confusion matrix but additionally the AUC-ROC, or true positive rate (TPR), vs. false positive rate (FPR), curve for the actual models used. Other Important Metrics include the

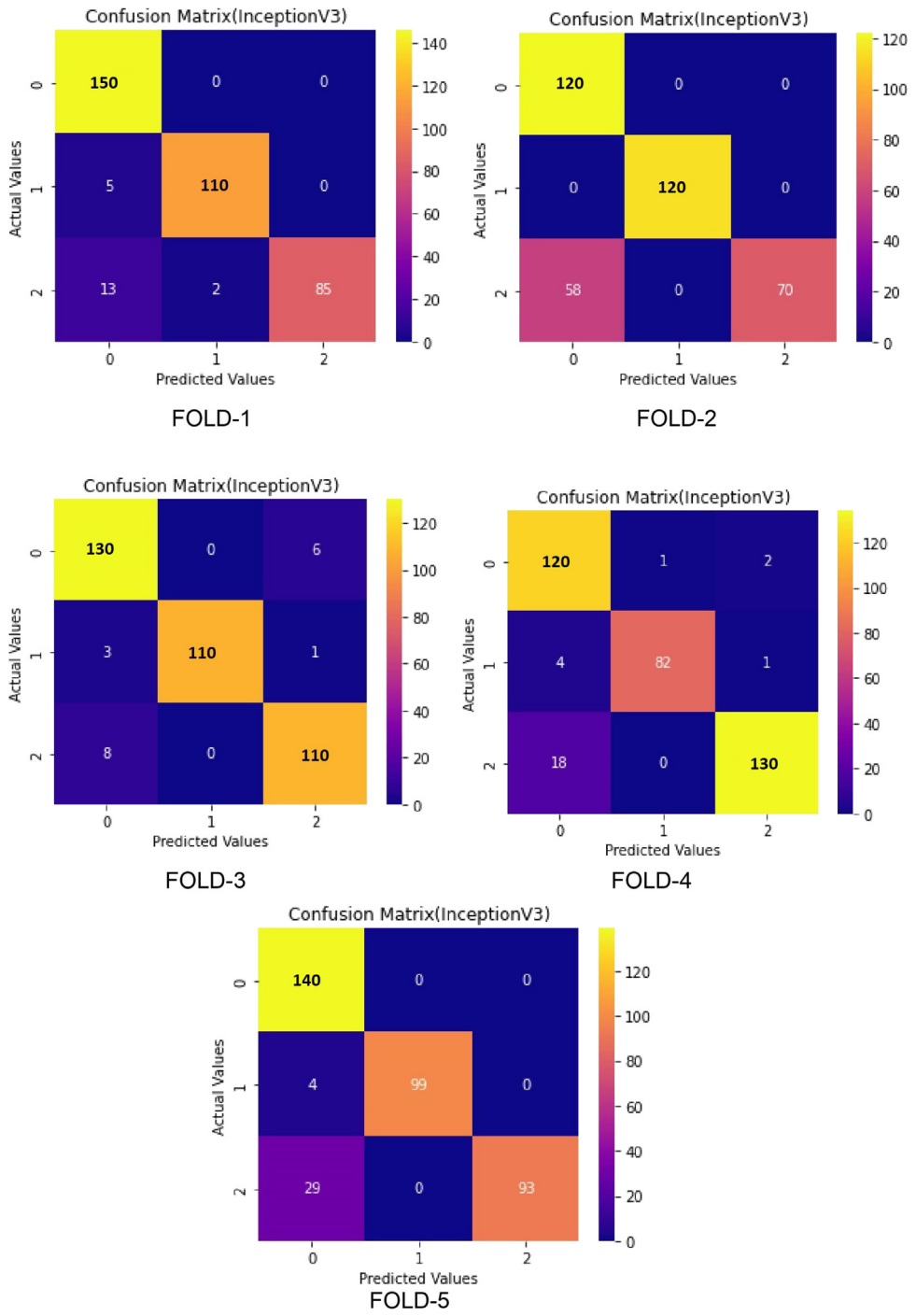

Fig. 5. Confusion Matrix:InceptionV3 on 5-Folds(0-Normal, 1-COVID19,2-Viral).

Accuracy versus Epoch Curve and the Loss versus Epoch curve, which are visualised in Fig. 9. From Fig. 8, it is analysed that the proposed algorithm has a very high true positive rate in comparison to a very high false positive rate, which signifies that the predicted results are mostly correct for a positive response, whereas Fig. 9 shows the loss and accuracy the model is having after each epoch to get an optimal epoch count.

4.4.2. Comparison of different datasets

The results of training and testing various transfer learning models on three different datasets are visualised in Fig. 10. To further evaluate the CoviDetector model, the best-performing model, InceptionV3 was evaluated on a small dataset after being trained on a separate dataset.

Moreover, the InceptionV3 model was trained on one Dataset in which 4 classes were converted into 3 and then the model was further tested on two other datasets containing 5 classes of data that was converted into 3 classes. Both the results are visualised in Figs. 12–14.

4.4.3. Semi-supervised ML

Apart from deep learning and transfer learning models, a semi-supervised method of machine learning was also implemented for image clustering [49–51]. In this paper, the K-means clustering model was implemented as a semi-supervised learning algorithm. The K-means clustering technique works on a specified number of clusters; in this scenario, the clusters varied from 2 to 20 different clusters. This method of getting the optimal number of clusters is known as the elbow

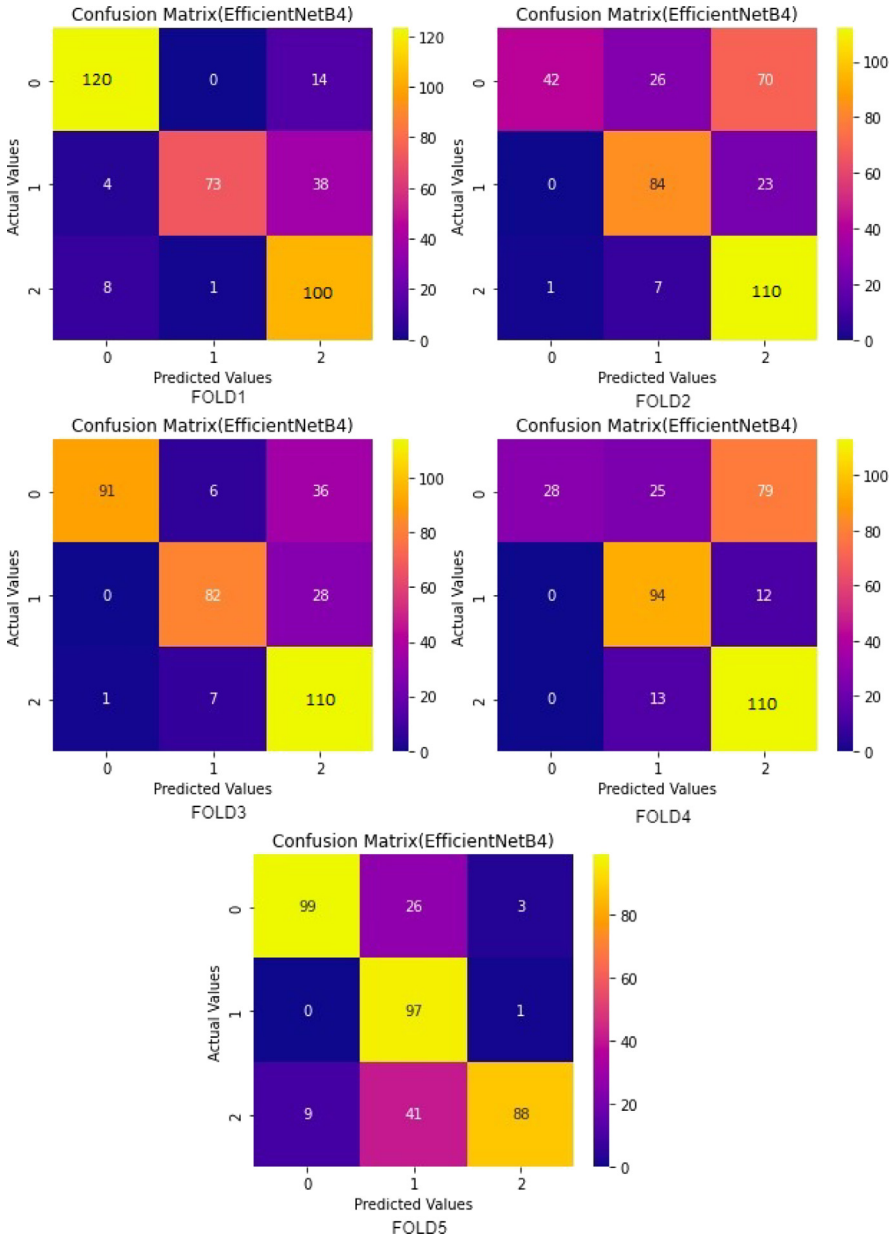

Fig. 6. Confusion Matrix:EfficientNetB4 on 5-Folds(0-Normal, 1-COVID19,2-Viral).

method. It was found that three clusters gave the best results out of all 20. The shuffled dataset was passed through the penultimate or second-last layer of the InceptionV3 functional model, and the extracted results were flattened and appended to a features variable. An instance of K-means clustering with three clusters was declared, and the features variable was fit using the same. The cluster labels were extracted and then compared with the original label data to compute the accuracy. In the semi-supervised method, we fed the training data through the best performing DNN i.e. Inception V3 and for each training dataset, we extracted the features of the penultimate layer. Then, k-means clustering was performed and k-clusters were extracted. The initial accuracy obtained was about 95% with the default hyperparameters when test data was passed through the clustering algorithm. A number of hyperparameters were tuned, like maximum iterations and tolerance, to further tweak the model. The final accuracy achieved for 3 classes of data is 99.69% when the clusters were plugged in with testing data. Fig. 11 shows the result for K-means clustering and the inception of V3-based semi-supervised learning on Dataset 1.

4.4.4. GradCAM

GradCAM is a form of post-hoc attention, meaning it is a method that has been devised for producing heatmaps by applying it to a pre-trained neural network model. The resultant effect is visual explanations from Deep Networks. The CXR image dataset has been used to train several deep learning models, namely VGG16, VGG19, InceptionV3, DenseNet and EfficientNet B4. As a result, all of them produced results with different levels of accuracy and precision. GradCAM has been used for a crystal clear visualisation of the results achieved from various models. GradCAM results have been laid out side by side in a comparative manner in Fig. 15.

Based on the data, we may infer that the suggested InceptionV3 model has superior accuracy and consistency. This was mostly due to the reduction of losses and the improvement in precision. The precision improved because we switched from using the Sequential CNN model to the transfer learning model. The InceptionV3 Transfer Learning Framework. We have also put the model into an Android application, which is not the case for the majority of transfer learning initiatives.

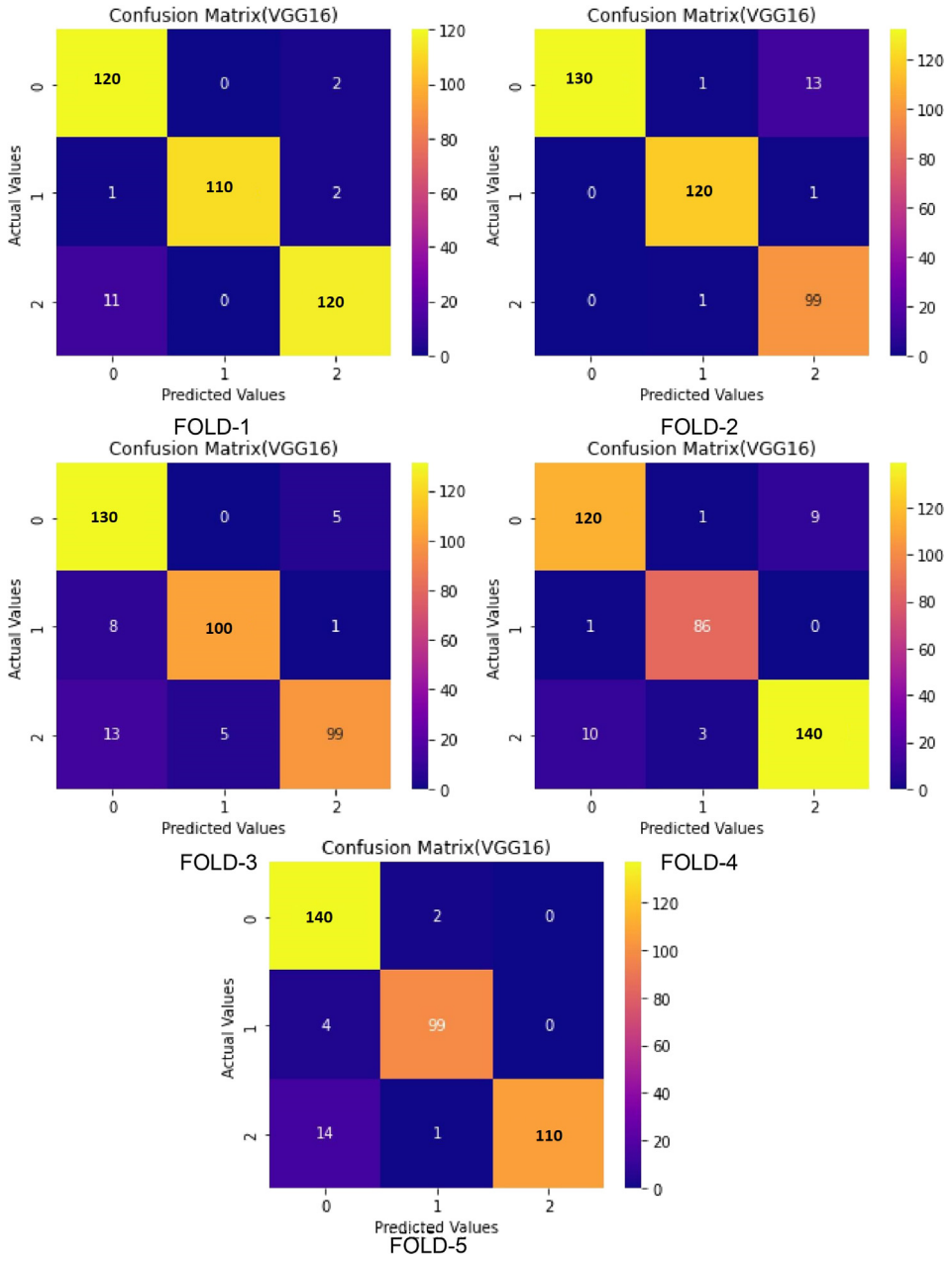

Fig. 7. Confusion Matrix:VGG16 on 5-Folds(0-Normal, 1-COVID19,2-Viral).

While all four transfer learning frameworks performed excellently, when comparing Accuracy and AUC Scores, the InceptionV3 model emerges on top. This is why InceptionV3 was chosen as the foundation for the Android app.

4.5. Cross validation

We used K-fold cross-validation and examined the different aspects of the data to ensure that the CoviDetector model does not overfit or underfit and works effectively. Due to the skewed nature of the data, K-fold (K=5) validation was necessary. At any given time, only one of the five sections of the dataset had been utilised for testing the model, while the others were utilised for training. We have performed the cross-validation on the best-performing model, i.e., InceptionV3. Fig. 16 shows the performance of the InceptionV3 model on various datasets. 5-fold cross-validation is performed for 3 class, 4 class, and 5 class classifications whose confusion matrix is shown in Figs. 12, 13 and 14 respectively.

5. Conclusions and future work

Accurate and timely detection of COVID-19 is necessary in today's world to prevent the further spread of this disease and timely treatment to start. In this study, we describe a method for quickly and easily identify COVID-19 positive patients. DNNs were shown to be effective at separating COVID-19 positive CXR pictures from Normal CXR images. In this paper, four techniques have been adopted, and the best overall has been selected for final classification. With the suggested model, we were able to attain a 99.65% accuracy in our classifications. As an added bonus, a specificity of 1.0 was attained. Pre-trained models can now be easily included in Android apps thanks to technological advancements. Therefore, we turned our focus to COVID-19 detection through Android smartphones. The proposed Android Application has a simple interface to browse through various images and upload one at a time. Once uploaded, the Android Application will be able to classify the image as a COVID-19 or Normal image.

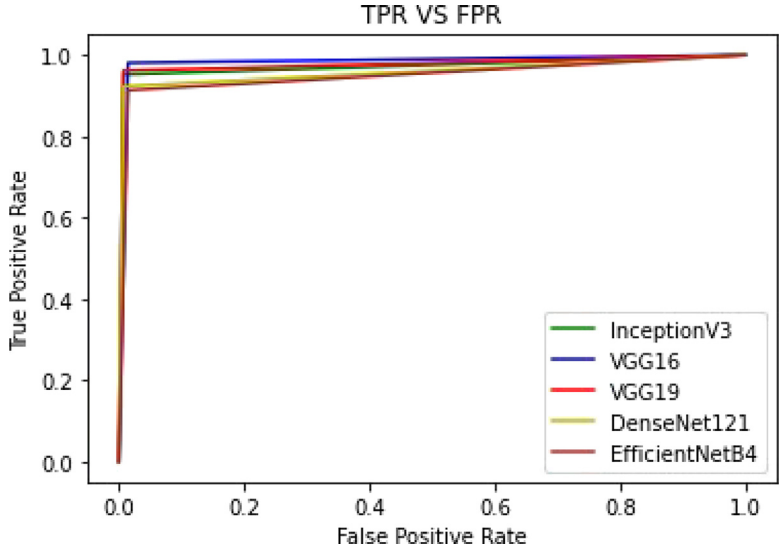

Fig. 8. AUC-ROC curve.

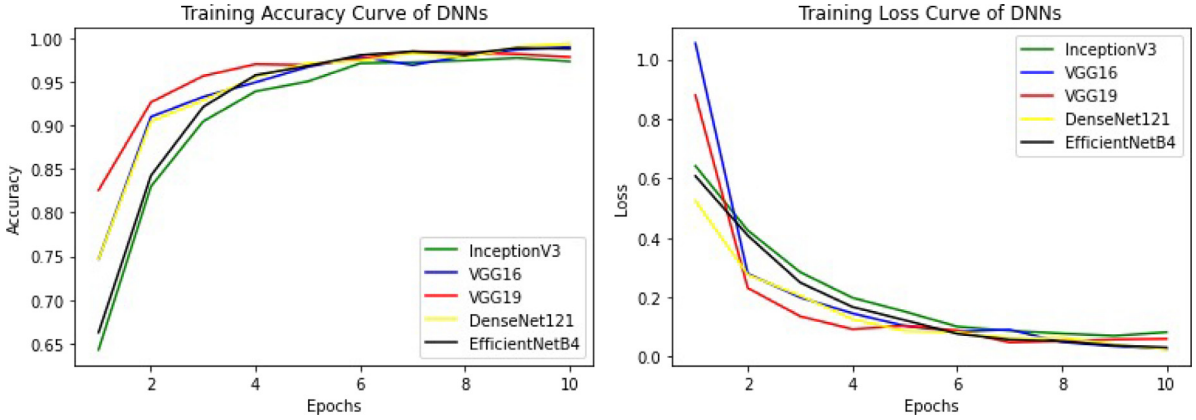

Fig. 9. Accuracy & Loss curves.

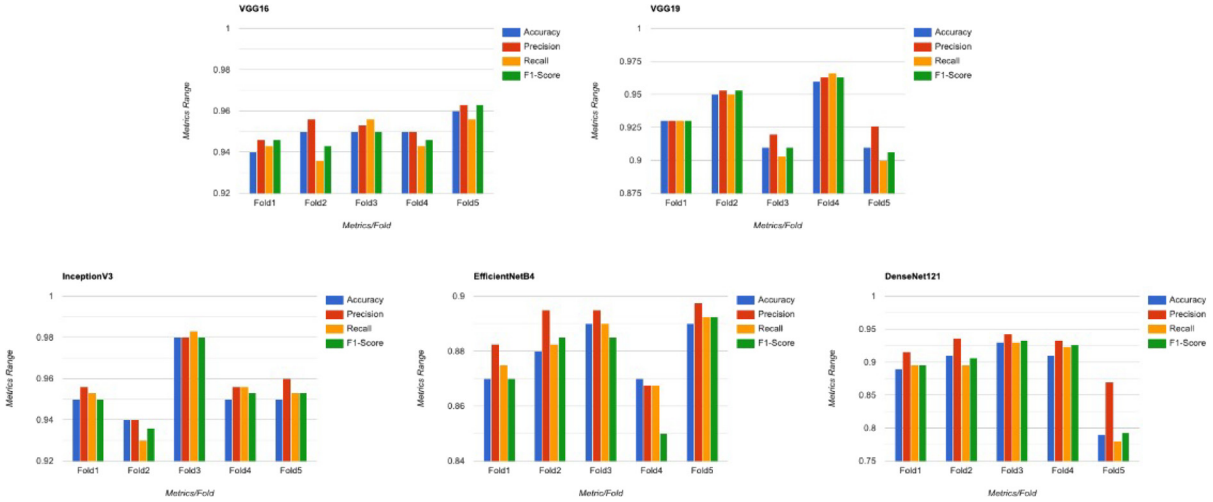

Fig. 10. Metrics chart for all models on Dataset1.

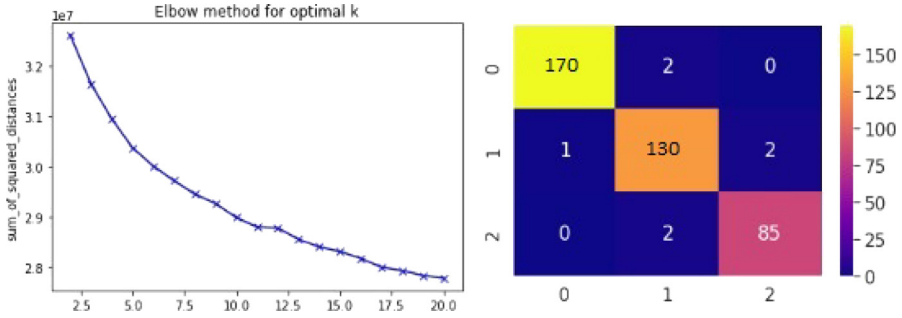

Fig. 11. Result for K-Means Clustering and Inception V3 based semi-supervised learning on Dataset1.

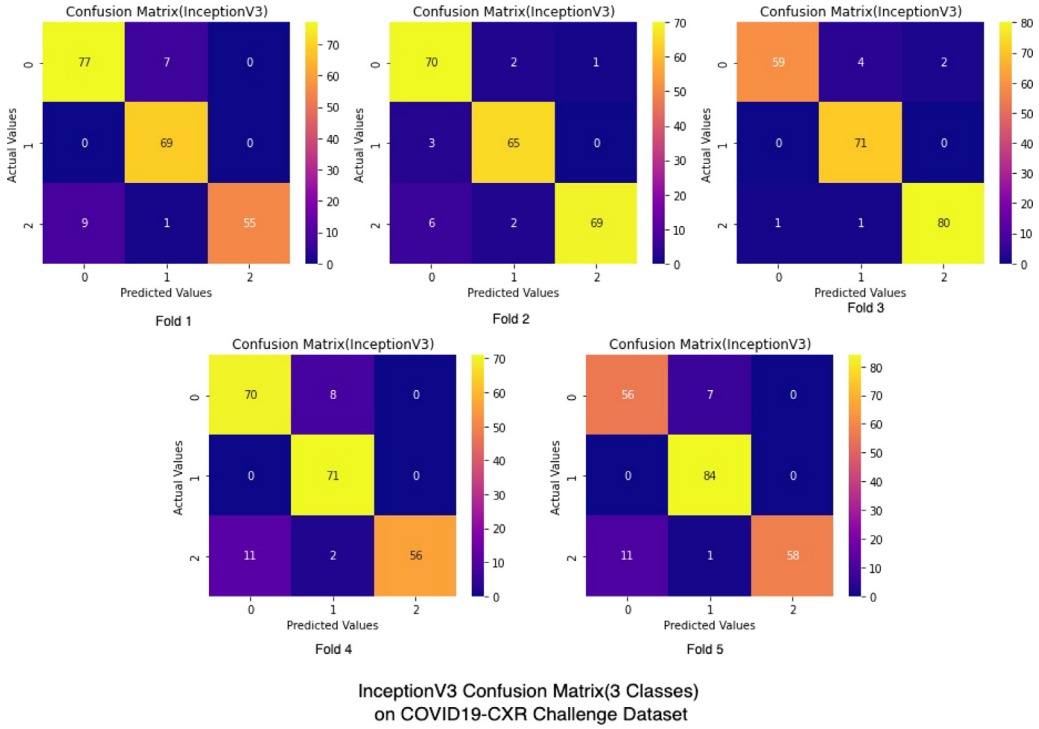

Fig. 12. InceptionV3 results on 3 class Dataset2.

5.1. Possible future directions

In future, CoviDetector can be extended in the following ways:

- **Internet of Things (IoT):** Deep learning algorithms proposed in this paper can be implemented in embedded devices such as the Raspberry Pi or Arduino and can be further used for building the smart X-ray-based COVID-19 detection [52]. CoviDetector also allows for the integration of ChatGPT and IoT, both of which may speed up and enhance patient care. Together, they form a formidable force that is altering the way we engage with technological advances and, perhaps, will make our lives better in generations and decades thereafter [53].
- **Web App:** Web application can be developed which will use this proposed deep learning framework in its backend for predicting the COVID-19 [54].
- **Artificial Intelligence (AI):** Advanced AI methods like quantum machine learning can be used to increase the accuracy rate of detection of COVID-19 [55].
- **Edge AI:** Since latency is a problem in mobile-based applications, we will utilise Edge AI in the future to develop an intelligent framework that will offload the latency-sensitive user requests to the edge node using the latest AI models without any further delay [56].
- **Security:** The CoviDetector itself has no built-in security protections, however, a cryptographic security mechanism might be added in the future to safeguard sensitive information [57].

Software availability

We released CoviDetector available for free as open source. All code, datasets, and result reproducibility scripts are publicly available and can be accessed from GitHub: https://github.com/dasanik2001/CoviDetector

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

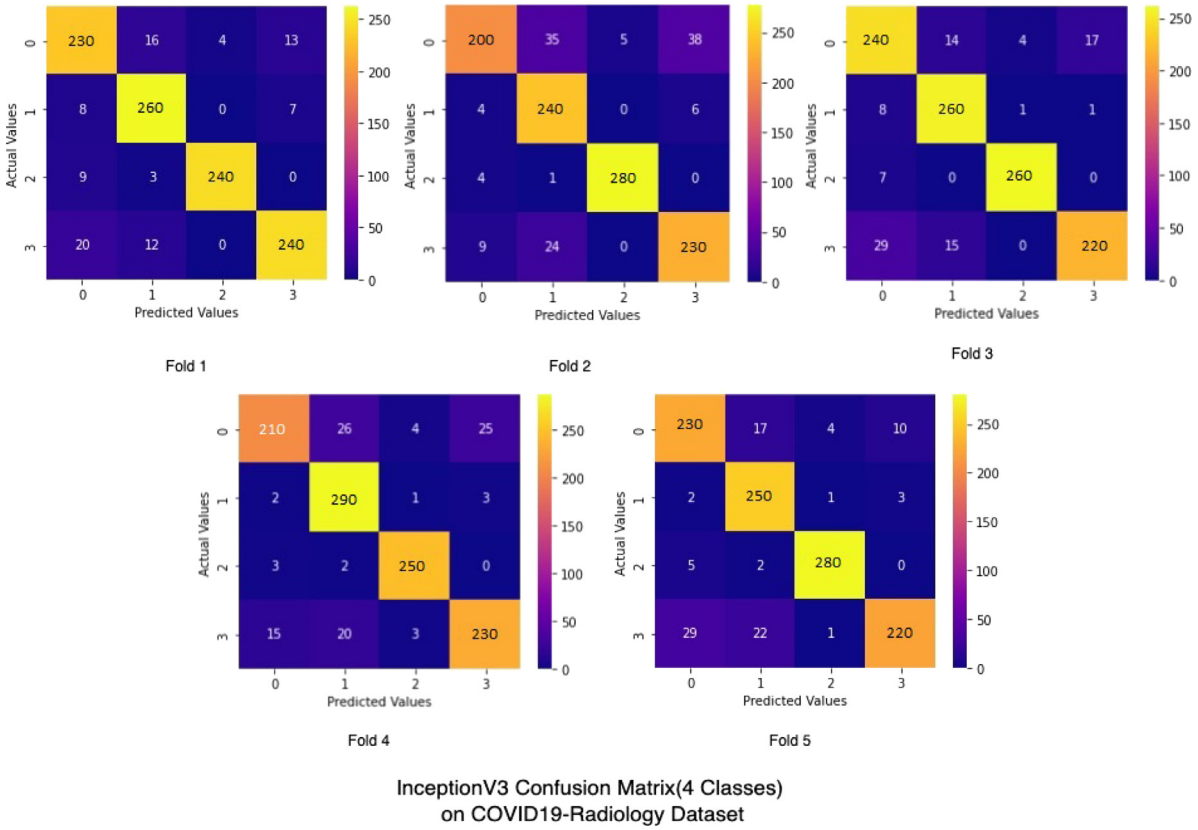

Fig. 13. InceptionV3 results on 4 class Dataset3.

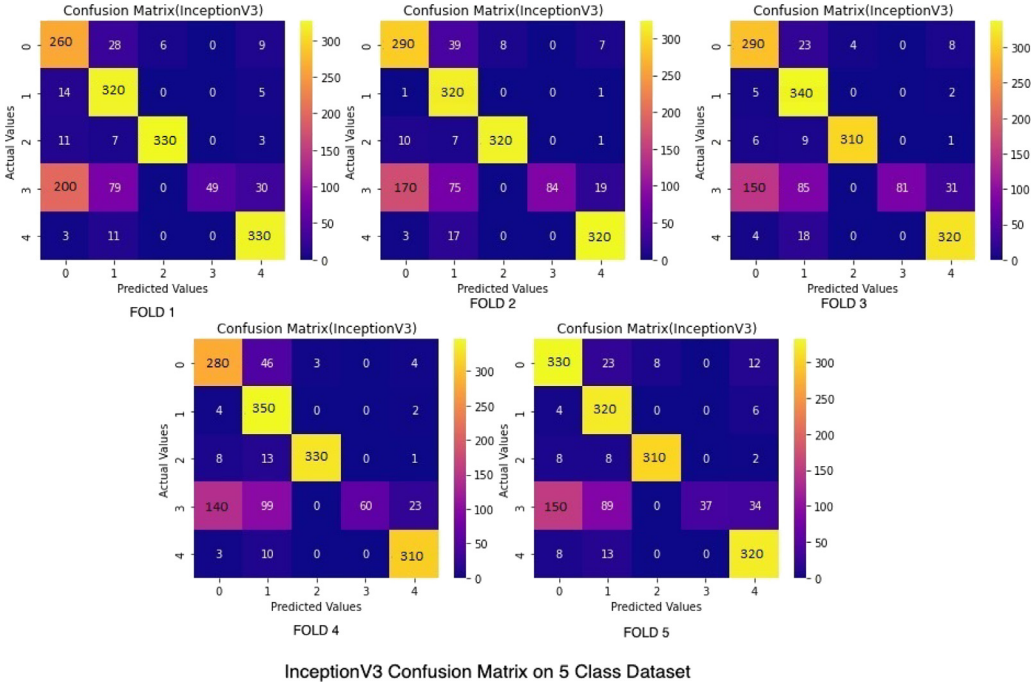

Fig. 14. InceptionV3 results on 5 class Dataset4.

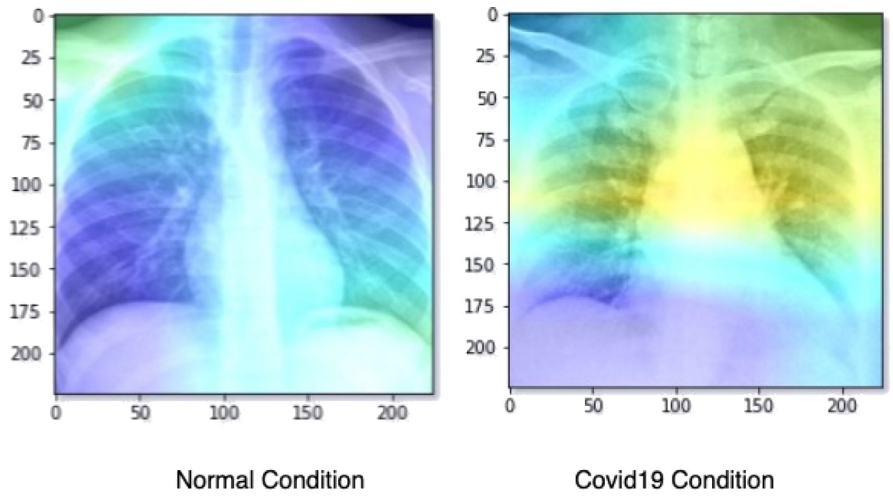

Fig. 15. GradCAM visualisation of InceptionV3-based model on COVID19 and normal condition CXR images.

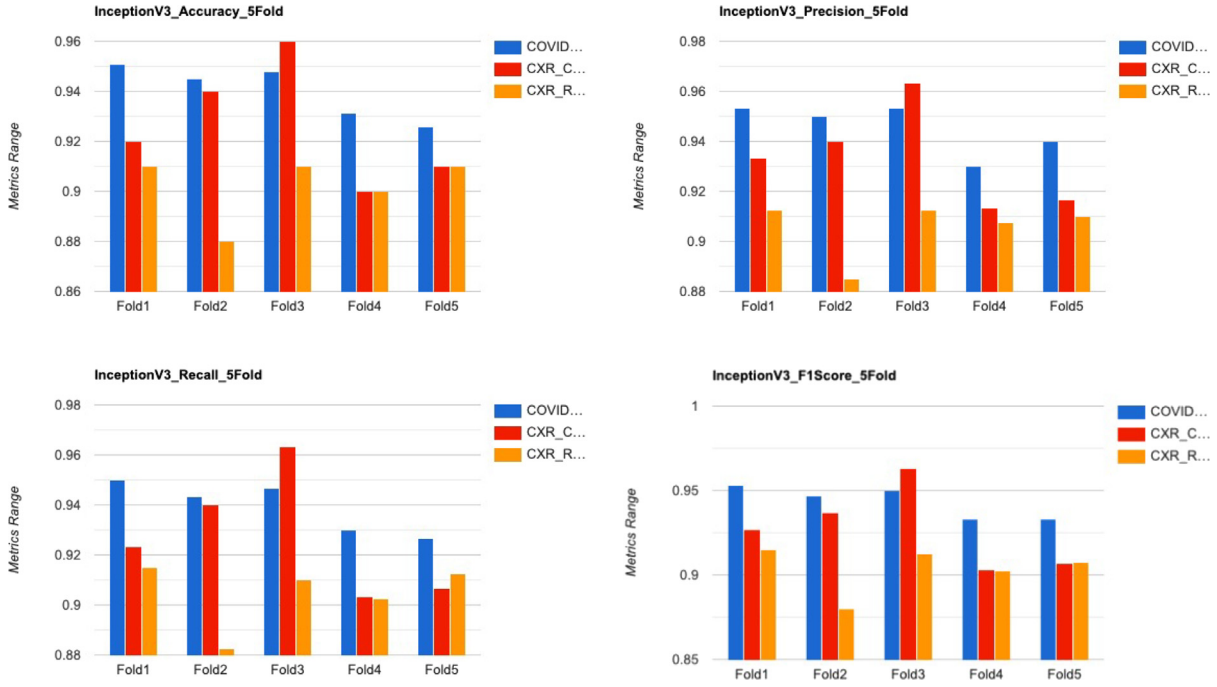

Fig. 16. Comparison of InceptionV3 on various datasets.

References

[1] Who Coronavirus (COVID-19) Dashboard, World Health Organization.

[2] Y. Huang, et al., Training, testing and benchmarking medical AI models using clinical aibench, BenchCouncil Trans. Benchmarks Stand. Eval. 2 (1) (2022) 100037.

[3] Omicron Variant: What You Need to Know, Centers for Disease Control and Prevention.

[4] F. Wu, et al., A new coronavirus associated with human respiratory disease in China, Nature 579 (2020) 1–8.

[5] A. Kumar, K. Sharma, et al., A drone-based networked system and methods for combating coronavirus disease (COVID-19) pandemic, Future Gener. Comput. Syst. 115 (2021) 1–19.

[6] M. Ahsan, et al., COVID-19 detection from chest X-ray images using feature fusion and deep learning, Sensors 21 (2021).

[7] A. Saygılı, A new approach for computer-aided detection of coronavirus (COVID-19) from CT and X-ray images using machine learning methods, Appl. Soft Comput. 105 (2021) 107323.

[8] A. Saygılı, Computer-aided detection of COVID-19 from CT images based on Gaussian mixture model and Kernel support vector machines classifier, Arab. J. Sci. Eng. 47 (2) (2022) 2435–2453.

[9] F. Desai, et al., HealthCloud: A system for monitoring health status of heart patients using machine learning and cloud computing, Internet Things 17 (2022) 100485.

[10] S. Tuli, et al., HealthFog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated IoT and fog computing environments, Future Gener. Comput. Syst. 104 (2020) 187–200.

[11] K. Bansal, et al., DeepBus: Machine learning based real time pothole detection system for smart transportation using IoT, Internet Technol. Lett. 3 (3) (2020) e156.

[12] S. Tuli, et al., Predicting the growth and trend of COVID-19 pandemic using machine learning and cloud computing, Internet Things 11 (2020) 100222.

[13] M.F. Aslan, et al., CNN-based transfer learning–BiLSTM network: A novel approach for COVID-19 infection detection, Appl. Soft Comput. 98 (2021) 106912.

[14] D. Kollias, A. Arsenos, S. Kollias, A deep neural architecture for harmonizing 3-D input data analysis and decision making in medical imaging, Neurocomputing 542 (2023) 126244.

[15] A. Arsenos, D. Kollias, S. Kollias, A large imaging database and novel deep neural architecture for COVID-19 diagnosis, in: 2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop, IVMSP, IEEE, 2022, pp. 1–5.

[16] I. Apostolopoulos, M. Tzani, COVID-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks, Australas. Phys.

Eng. Sci. Med. 43 (2020) Supported By the Australasian College of Physical Scientists in Medicine and the Australasian Association of Physical Sciences in Medicine.
[17] P. Verma, et al., FCMCPS-COVID: AI propelled fog–cloud inspired scalable medical cyber-physical system, specific to coronavirus disease, Internet Things 23 (2023) 100828.
[18] M. Golec, et al., HealthFaaS: AI based smart healthcare system for heart patients using serverless computing, IEEE Internet Things J. (2023).
[19] M. Singh, et al., Quantifying COVID-19 enforced global changes in atmospheric pollutants using cloud computing based remote sensing, Remote Sens. Appl. Soc. Environ. 22 (2021) 100489.
[20] M.M. Islam, et al., Diagnosis of COVID-19 from X-rays using combined CNN-RNN architecture with transfer learning, BenchCouncil Trans. Benchmarks Stand. Eval. 2 (4) (2022) 100088.
[21] M. Golec, et al., IFaaSBus: A security-and privacy-based lightweight framework for serverless computing using IoT and machine learning, IEEE Trans. Ind. Inform. 18 (5) (2021) 3522–3529.
[22] D. Mittal, A deep learning approach to detect COVID-19 coronavirus with X-ray images, Biocybern. Biomed. Eng. 40 (2020).
[23] F. Ucar, D. Korkmaz, Covidiagnosis-net: Deep Bayes-SqueezeNet based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images, Med. Hypotheses 140 (2020) 109761.
[24] K. Ahammed, et al., Early detection of coronavirus cases using chest X-ray images employing machine learning and deep learning approaches, 2020.
[25] M. Azemin, et al., COVID-19 deep learning prediction model using publicly available radiologist-adjudicated chest X-Ray images as training data: Preliminary findings, Int. J. Biomed. Imaging 2020 (2020) 1–7.
[26] T. Ozturk, et al., Automated detection of COVID-19 cases using deep neural networks with X-ray images, Comput. Biol. Med. 121 (2020).
[27] I. Khan, N. Aslam, A deep-learning-based framework for automated diagnosis of COVID-19 using X-ray images, Information 11 (2020) 419.
[28] X. Wang, et al., A weakly-supervised framework for COVID-19 classification and lesion localization from chest CT, IEEE Trans. Med. Imaging PP (2020) 1.
[29] Y. Oh, S. Park, J. Ye, Deep learning COVID-19 features on CXR using limited training data sets, IEEE Trans. Med. Imaging PP (2020) 1.
[30] M. Mohamed, et al., COVID-19 detection from chest X-ray images using artificial-intelligence-based model imported in a mobile application, 2021.
[31] K. Bushra, et al., Automated detection of COVID-19 from X-ray images using CNN and android mobile, Res. Biomed. Eng. 37 (2021).
[32] M. Taresh, et al., Transfer learning to detect COVID-19 automatically from X-ray images using convolutional neural networks, Int. J. Biomed. Imaging 2021 (2021) 1–9.
[33] M. Ahsan, et al., COVID-19 detection from chest X-ray images using feature fusion and deep learning, Sensors 21 (2021).
[34] D.-P. Fan, et al., Inf-Net: Automatic COVID-19 lung infection segmentation from CT scans, 2020.
[35] M. Loey, et al., Within the lack of chest COVID-19 X-ray dataset: A novel detection model based on GAN and deep transfer learning, Symmetry 12 (2020) 651.
[36] N. Wang, et al., Deep learning for the detection of COVID-19 using transfer learning and model integration, 2020, pp. 281–284.
[37] T. Mahmud, et al., CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization, Comput. Biol. Med. 122 (2020) 103869.
[38] S. Minaee, et al., Deep-COVID: Predicting COVID-19 from chest X-ray images using deep transfer learning, Med. Image Anal. 65 (2020) 101794.
[39] S. Tabik, et al., COVIDGR dataset and COVID-sdnet methodology for predicting COVID-19 based on chest X-ray images, IEEE J. Biomed. Health Inf. 24 (2020) 3595–3605.
[40] J. Xiao, et al., Application of a novel and improved VGG-19 network in the detection of workers wearing masks, J. Phys. Conf. Ser. 1518 (2020) 012041.
[41] S. Tammina, Transfer learning using VGG-16 with deep convolutional neural network for classifying images, Int. J. Sci. Res. Publ. (IJSRP) 9 (2019) p9420.
[42] S. Wang, Y.-D. Zhang, DenseNet-201-based deep neural network with composite learning factor and precomputation for multiple sclerosis classification, ACM Trans. Multimed. Comput. Commun. Appl. 16 (2020) 1–19.
[43] K. Boonyuen, et al., Convolutional neural network inception-v3: A machine learning approach for leveling short-range rainfall forecast model from satellite image, 2019, pp. 105–115.
[44] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.
[45] M. Siddhartha, A. Santra, COVIDLite: A depth-wise separable deep neural network with white balance and CLAHE for detection of COVID-19, 2020.
[46] D. Kermany, et al., Identifying medical diagnoses and treatable diseases by image-based deep learning, Cell 172 (2018) 1122–1131.e9.
[47] A. Tahir, et al., COVID-19 infection localization and severity grading from chest X-ray images, 2021.
[48] M. Chowdhury, et al., Can AI help in screening Viral and COVID-19 pneumonia? IEEE Access 8 (2020) 132665–132676.
[49] D. Kollias, A. Tagaris, A. Stafylopatis, S. Kollias, G. Tagaris, Deep neural architectures for prediction in healthcare, Complex Intell. Syst. 4 (2018) 119–131.
[50] N. Bouas, Y. Vlaxos, V. Brillakis, M. Seferis, S. Kollias, Deep transparent prediction through latent representation analysis, 2020, arXiv preprint arXiv: 2009.07044.
[51] Y. Vlaxos, M. Seferis, S. Kollias, Transparent adaptation in deep medical image diagnosis, in: Trustworthy AI-Integrating Learning, Optimization and Reasoning: First International Workshop, TAILOR 2020, Virtual Event, September 4–5, 2020, Revised Selected Papers 1, Springer, 2021, pp. 251–267.
[52] T. Shao, et al., IoT-Pi: A machine learning-based lightweight framework for cost-effective distributed computing using IoT, Internet Technol. Lett. (2022) e355.
[53] S.S. Gill, R. Kaur, ChatGPT: Vision and challenges, Internet Things Cyber-Phys. Syst. 3 (2023) 262–271.
[54] D. Chowdhury, et al., Covacdiser: A machine learning-based web application to recommend the prioritization of COVID-19 vaccination, in: Intelligence Enabled Research, Springer, 2022, pp. 105–117.
[55] S.S. Gill, et al., AI for next generation computing: Emerging trends and future directions, Internet Things (2022) 100514.
[56] R. Singh, et al., Edge AI: a survey, Internet Things Cyber-Phys. Syst. 3 (2023).
[57] Y. Zhou, et al., An efficient encrypted deduplication scheme with security-enhanced proof of ownership in edge computing, BenchCouncil Trans. Benchmarks Stand. Eval. 2 (2) (2022) 100062.