


RESEARCH

Open Access



The use of Efficientnet_b0 network to identify COVID-19 in chest X-ray images

Michael Ayitey Junior^{1*} , Peter Appiahene², Yaw Marfo Missah³ and Vivian Akoto-Adjepong²

*Correspondence:
michaelayiteyjunior@gmail.com

¹Information Technology and Decision Sciences, University of Energy and Natural Resources, Sunyani, Ghana

²Department of Computer Science and Informatics School of Sciences, University of Energy and Natural Resources, Sunyani, Ghana

³Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

Abstract

A newly discovered coronavirus called COVID-19 poses the greatest threat to mankind in the twenty-first century. Mortality has dramatically increased in all cities and countries due to the virus's current rate of spread. A speedy and precise diagnosis is also necessary in order to treat the illness. This study identified three groups for chest X-ray images: Covid, normal, and pneumonia. This study's objective is to present a framework for categorizing chest X-ray images into three groups of pneumonia, normal, and Covid scenarios. To do this, chest X-ray images from the Kaggle database which have been utilized in previous studies were obtained. It is suggested to use an Efficientnet_b0 model to identify characteristics in raw data hierarchically. An unedited X-ray image of the chest is enhanced for more reasonable assumptions in order to apply the proposed method in real-world situations. With an overall accuracy of 93.75%, the proposed network correctly identified the chest X-ray images to the classes of Covid, viral pneumonia, and normal on the test set. 90% accuracy rate for the test dataset was attained for the viral pneumonitis group. On the test dataset, the Normal class accuracy was 94.7%, while the Covid class accuracy was 96%. The findings indicate that the network is robust. In addition, when compared to the most advanced techniques of identifying pneumonia, the concluded findings from the suggested model are highly encouraging. Since the recommended network is successful at doing so utilizing chest X-ray imaging, radiologists can diagnose COVID-19 and other lung infectious infections promptly and correctly.

Keywords: Coronavirus, X-ray, Convolutional neural network, Efficientnet_b0

Introduction

Globally, COVID-19 had a profound effect on people and healthcare systems [19]. The World Health Organization (WHO) declared the coronavirus infection (COVID-19) as a worldwide pandemic. The virus was discovered for the first time in mainland China's Hubei province's capital city Wuhan in December 2019 [1]. People who are near together will spread COVID-19 more quickly. According to data from John Hopkins University, there were 641,880,840 fatalities and 584,884,407 illnesses worldwide as of July 29, 2022. Restrictions on movement assist in keeping the sickness under control, and careful hand washing is always advised to avoid viral infections. Meanwhile, the most common signs of illness are fever and cough. Chest ache, a sore throat, and phlegm coughing are possible additional symptoms. Viral pneumonia,

which carries a 5.8% mortality risk, may lead to COVID-19. The COVID-19 mortality rate is 5% that of the Spanish flu epidemic in 1918. The following vaccines currently have Emergency Use Listings as of January 12, 2022. (EUL): Pfizer/BioNTech Comirnaty vaccine, SII/COVISHIELD and AstraZeneca/AZD1222 vaccines, The Janssen/Ad26.COV 2.S vaccine developed by Johnson & Johnson, Moderna COVID-19 vaccine (mRNA 1273), Sinopharm COVID-19 vaccine, Sinovac-CoronaVac vaccine, Bharat Biotech BBV152 COVAXIN vaccine, The Covovax (NVX-CoV2373) vaccine, The Nuvaxovid (NVX-CoV2373) vaccine to be use to vaccinate people against the disease. The virus's genomic sequence demonstrates that its DNA changes to enable it to live in each environment it encounters, and Fig. 1 depicts what the virus looks like under a microscope.

COVID-19 is often diagnosed using reverse transcriptase-polymerase chain reaction (RT-PCR), a laboratory process that interreact with other ribonucleic (RNA) and deoxyribonucleic (DNA) acids to quantify the quantity of certain ribonucleic acids using fluorescence [3]. Nasal secretion research samples are subjected to RT-PCR analysis. To get the samples, a swab is softly placed into the nose and pushed into the nasopharynx to collect secretions. Even though it is currently expensive, sluggish, and in great demand, it gives a high level of accuracy. Despite being often available,

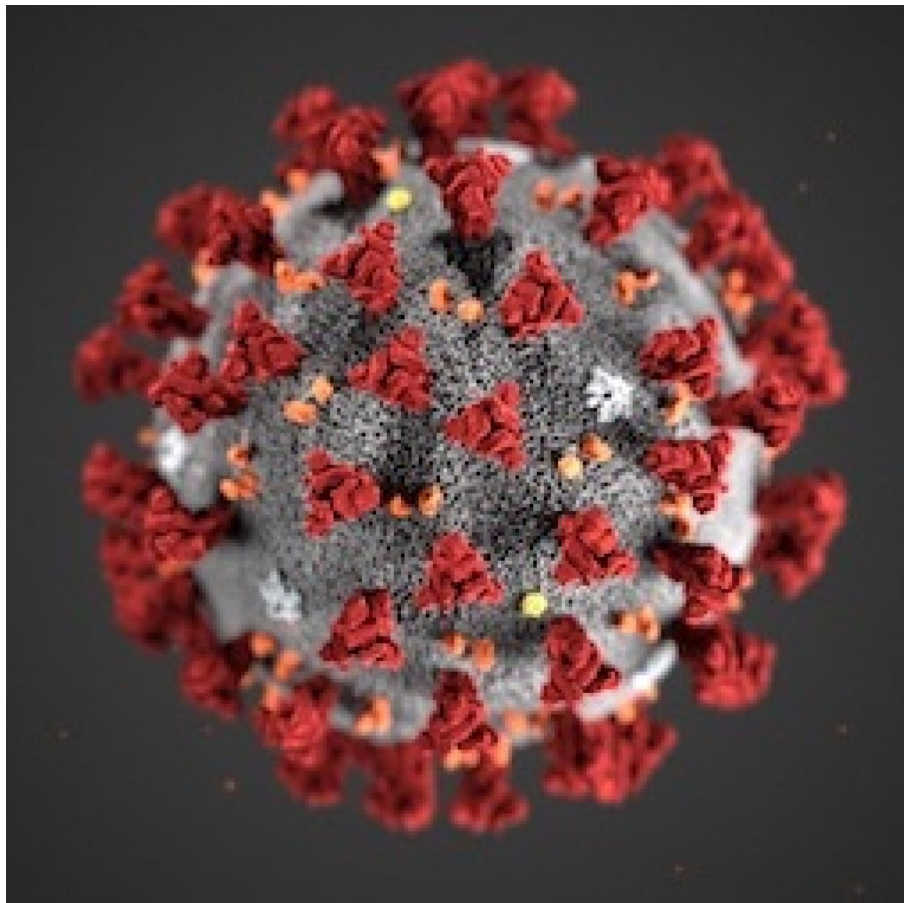


Fig. 1 Ultrastructural morphology of the coronavirus as seen under an electronic microscope [6] (COVID-19)—Images | BMJ Best Practice)

faster, and less costly, chest computed tomography (CT) scans and chest X-rays might be difficult to interpret for the signals that signify the presence of COVID-19 in the lungs.

RT-PCR was less effective than CT scans and X-rays for detecting COVID-19 symptoms in the lower respiratory tract [15]. In rare circumstances, RT-PCR testing can take the role of CT scans and X-ray examinations. This will be possible because to the development of advanced diagnostic systems employing artificial intelligence (AI) methods. CT scans and chest X-ray images are utilized to detect COVID-19 in a person more quickly and easily.

In recent years, healthcare prediction has played a major role in saving lives [2]. An initial COVID-19 patient examination can be performed using AI as an alternative to labor- and time-intensive conventional techniques. In this work, we highly advocate the use of AI to forecast COVID-19 occurrences and identify COVID-19 patients using chest X-ray images. The objective of this study is to categorize chest X-ray images into Covid, normal, and pneumonia groups; as a result, this is a multiclass classification issue. We consider CNN-based architecture in this situation since it is known for its better acceptance performance in image categorization or detection.

These are the primary contributions of the work:

- To categorize chest X-ray images into Covid, normal, and pneumonia using convolutional neural network-based architecture which incorporates back-end pre-trained EfficientNet_b0 for feature extraction and model snapshots.
- Empirical results from 317 patient chest X-ray images show that Efficientnet_b0 model has a 93–98% accuracy rate for detecting COVID-19 patients.

The rest of the article is organized as follows: related studies about COVID-19 detection are presented at Section 2. Section 3 presents the whole model architecture, dataset description, and performance-evaluation metrics. Section 4 presents the results and discussions in that order.

Related studies

Since 2020, there have been several investigations toward the analysis and detection of COVID-19 [9]. This section's first portion discusses problems with COVID-19 identification using chest X-ray and CT scan data combined with deep learning techniques. The second section examines pertinent literature to evaluate projected projections for the number of COVID-19 confirmations.

The rapid global spread of COVID-19 has caused it to become a pandemic. It is quite challenging to locate affected persons since they do not instantly show symptoms of illness. Therefore, a routine method of counting the number of potentially impacted persons is needed in order to take the appropriate action. The traditional method of an individual's COVID-19 assessment which is expensive and time consuming can be alternatively replace with artificial intelligence process. This study focused on using Deep Vision to detect COVID-19 infections in patients and forecast COVID-19 infections using chest X-ray images, despite the fact that COVID-19 has been the topic of multiple research.

Numerous studies utilized AI to predict the sickness since the COVID-19 epidemic [7, 16]. AI's primary benefit is that it can be used to train models to categorize hidden images. In this study, chest X-ray images were analyzed using AI to determine if patients had COVID-19, healthy pneumonia, or viral pneumonia.

COVID-19 diagnosis using artificial intelligence (deep learning)

A modern machine learning system built on a convolutional neural network (CNN) is called the deep learning system made its debut in 2012. Computing models with several processing layers may learn data representation by moving through various abstraction levels using deep learning techniques. To do categorization tasks from texts, images, or sounds, they build a computer model.

Using a sample of 2000 radiograms, transfer learning was utilized to train four well-known convolutional neural networks, ResNet18, ResNet50, SqueezeNet, and DenseNet-121, to detect COVID-19 illness in the inspected chest X-ray images. When these models were evaluated on 3000 images, the bulk of these networks had a sensitivity rate of 98%, while having a specificity rate of roughly 90% [13]. Deep-learning models, according to Alazab et al. [1], have high accuracy and even occasionally improve human performance. X-ray machines employ radiation in the form of light or radio waves to examine body parts for signs of cancer, lung disease, fractures, and other injuries. While this is going on, modern X-ray technology called CT scans is being used to examine the soft tissues and organs of active biological components.

Islam et al. [11] applied a deep CNN-LSTM network to a dataset of 4575 X-ray images, including 1525 images of COVID-19, to identify a new coronavirus (COVID-19) using X-ray images. The suggested method obtained accuracy of 99.4%, AUC of 99.9%, specificity of 99.2%, sensitivity of 99.3%, and F1-score of 98.9%, according to the testing data. With the help of the currently accessible dataset, the system produced the expected results. The system will operate much more efficiently if more COVID-19 images are made available. The recommended strategy aids physicians in accurately diagnosing and treating COVID-19 patients.

In a related research, Bassi and Attux [3] extracted characteristics from chest X-ray images and classified the images as COVID-19, pneumonia, or normal using dense convolutional networks with transfer learning. They were able to attain an accuracy rate of 100% on our test dataset.

Furthermore [15] suggested an automated detection technique for the diagnosis of patients with coronavirus pneumonia utilizing chest X-ray radiographs and five convolutional neural network-based models that had been previously trained (ResNet50, ResNet101, ResNet152, InceptionV3, and Inception-ResNetV2). They created three separate binary classifications with four classes (COVID-19, normal (healthy), viral pneumonia, and bacterial pneumonia) using fivefold cross-validation. Compared to the other four models, the pre-trained ResNet50 model outperforms them in terms of classification accuracy (96.1% accuracy for Dataset-1, 99.5% accuracy for Dataset-2, and 99.7% accuracy for Dataset-3).

A model for the automatic prediction of COVID-19 from chest images was also created by [9] using a modified version of ResNet50. The modification was made by including three extra layers with the labels Conv, Batch Normalize, and Activation Relu. The

ResNet50 design incorporates these layers for trustworthy feature extraction and precise discrimination. According to test results, the suggested modification, injected layers, improves upon current methods by raising diagnostic accuracy to 97.7% for Computerized Tomography dataset and 97.1% for X-Ray dataset.

According to [18], early identification and detection are essential for stopping the spread of COVID-19. Several deep learning-based approaches for COVID-19 screening in CT scans have recently been developed to automate and aid in the diagnosis. However, one or more of the following problems exists with these techniques: (i) They treated every CT scan slice separately, (ii) they used datasets with identical sets of images to test and train the algorithms. The same patient can appear concurrently in the training and test sets if the slices are treated individually, which could lead to inaccurate findings.

Furthermore, it calls into question the wisdom of comparing scans from the same patient at the same time. Furthermore, the use of a specific dataset raises concerns about the generalizability of the approaches. The quality of the images from various CT scanners and databases may vary, reflecting the conditions in the nations and regions from which they were obtained. In their work, they offered an efficient deep learning approach that merged a voting-based approach with COVID-19 screening to solve these two issues. Using a voting method, the approach grouped the images from a certain patient. To evaluate the method, the two largest datasets for COVID-19 CT analysis were split based on the patients. To evaluate the models' resilience in a more real-world setting where data originate from various distributions, a cross-dataset analysis is also offered. As accuracy drops from 87.68 to 56.16% in the best evaluation scenario, the cross-dataset study has shown that deep learning models' generalization power is woefully inadequate for the task at hand. Their research demonstrated that methods for COVID-19 detection using CT scans require substantial advancement prior to getting considered as a therapy option. To evaluate the approaches in a real-world situation, larger and more diverse datasets are also needed.

Materials and methods

The steps of our proposed deep learning-based COVID-19 detection are shown in Fig. 6. The following five steps represent a summary of the phases: methods in a practical setting.

Step 1: Dataset gathering and preparation.

Step 2: Data augmentation.

Step 3: Implement deep learning (Efficientnet_b0).

Step 4: On the test dataset, analyze the model's effectiveness.

Dataset

Due to anonymity and confidentiality the dataset has no names attached; Fig. 2 is a random selection of some images from the dataset. The dataset included the following information: (a) A training dataset comprising chest X-ray images of three patient classes of Covid, normal, and viral pneumonia patients. (b) The test dataset containing chest X-ray images also with three classes of Covid, normal, and viral pneumonia patients. The original dataset was obtained from the Kaggle database with 317 total number of images and breakdown in Table 1 as shown below.

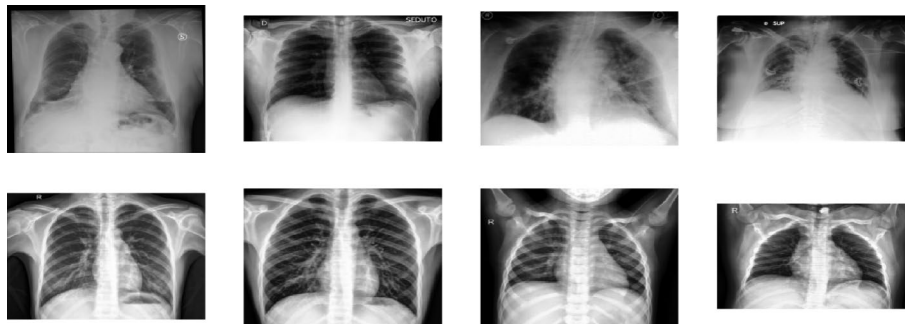


Fig. 2 Sample X-ray images of patients

Table 1 Original dataset and its splitting from Kaggle database

	Classes	Number		Classes	Number
Training dataset	Covid	111	Test dataset	Covid	26
	Normal	70		Normal	20
	Viral pneumonia	70		Viral pneumonia	20

```

+-----+
| NVIDIA-SMI 460.32.03      Driver Version: 460.32.03      CUDA Version: 11.2      |
+-----+-----+-----+
| GPU  Name      Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|====+=====+====+=====+=====+=====+=====+=====+
|  0  Tesla T4           Off      | 00000000:00:04:0 Off |    0%      Default  |
| N/A   40C   P8      10W / 70W |  0MiB / 15109MiB |      0%      Default  |
|                               |                      | N/A         N/A     |
+-----+-----+-----+

+-----+
| Processes: |
| GPU  GI  CI           PID  Type  Process name                        GPU Memory |
| ID   ID  ID           |  |      |                                     Usage      |
+-----+-----+-----+
| No running processes found |
+-----+
    
```

Fig. 3 image showing the type of GPU available on Colab for running the model

Model

Implementation environment

Google’s open-source and online machine learning platform Colab is used for a variety of tasks, including research that necessitates a sizable computing infrastructure and a large collection of machine learning libraries. It enables machine learning experts to quickly design and launch Python applications on Google’s cloud servers. This platform is a wonderful match for our project thanks to features like easy sharing and free access to internet graphics processing units (GPUs). Making the choice even simpler, it integrates the significant machine learning languages listed below. Torchvision, a package for computer vision, was already included in Colab and was not installed; nevertheless, the program did install the image timm library. The characteristics of the kind of GPU made accessible on Google Colab for using the model are shown in Fig. 3.

Image preprocessing

Two steps are involved in the image preprocessing:

Step1: The least height and breadth of the dataset images were first determined by comparing all of the initially obtained images. All of the images in the collection were scaled to 224×224 when this lowest dimension was found.

Step2: Pre-processing is done on resized images in line with the ImageNet database. Millions of images with tens of thousands of distinct image classifications are part of the publicly available ImageNet computer vision dataset. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) encourages the creation of cutting-edge algorithms by utilizing a portion of the ImageNet database's collection of images [10]. In this study, a network that has previously been trained on the ImageNet database is used to train the network on our dataset. Transfer learning is the technique of employing a network that has already been trained on the ImageNet database to train a network on a new dataset. The scaled images from step 1 are next preprocessed in line with the images in the ImageNet database. Before subtracting the ImageNet mean and dividing the ImageNet standard deviation, each pixel in the ImageNet database is divided by 255. The images in our dataset are handled uniformly. The following ImageNet statistics are used to individually normalize each of the three channels of an image's pixels: Channels 1, 2, and 3's respective means are 0.48, 0.456, and 0.406 correspondingly. The standard deviations for Channels 1, 2, and 3 were 0.229, 0.224, and 0.225, respectively.

Final shape and input size of images: we used a four-dimensional data loader tensor of batch, channel, height and weight Number of batches: 32.

Number of channels: 3

Height of image = 224.

Weight image = 224.

78% of the X-rays from the train images were used to produce the training dataset, while 22% were used to create the test dataset (holdout).

Data augmentation

A group of methods referred to as data augmentation are used to produce extra data points from the present data in order to artificially enhance the amount of data. Deep learning models can be used to create new data or make tiny alterations to existing data. Machine learning models become more robust as a result of data augmentation techniques that simulate variations that will be met by the models in a real situation. In this investigation, the data augmentation techniques employed were cropping and random erasing.

(I) Center cropping: This layer crops the center of the images to a 224 by 224 target size. An image will be enlarged and cropped if it is lower than the goal size in order to return the biggest window that still maintains the desired aspect ratio. (II) By selecting a rectangular section in an image, random values are applied to the pixels that will be erased [21]. Random erasing is simple to use, parameter learning-free, and compatible with the majority of CNN-based recognition models. Now that all the augmentation are applied, Fig. 4 shows a plot of some of those augmented images.

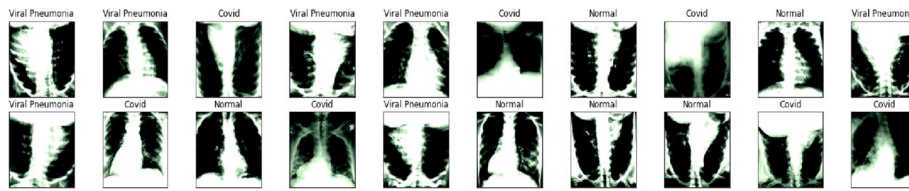


Fig. 4 Sample augmented images of the dataset

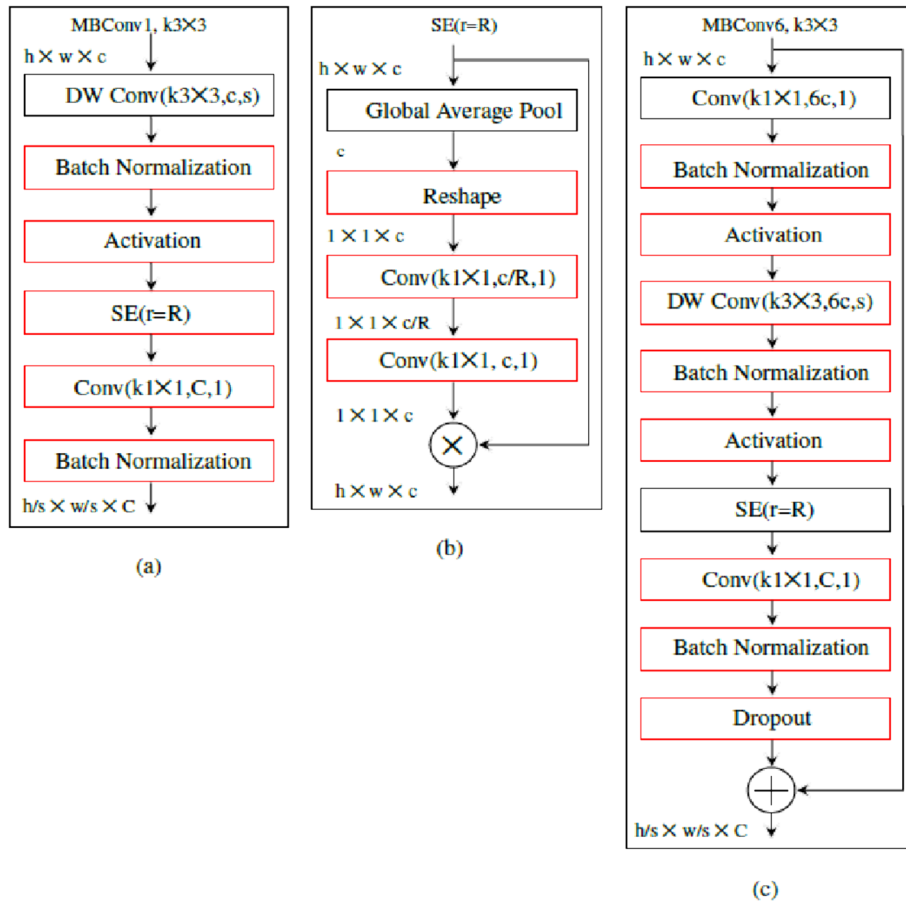


Fig. 5 EfficientNet_b0 architecture

Efficientnet_b0 network architecture

In this part, we will provide a brief introduction of the Efficientnet_b0 network architecture. After enhancing the COVID dataset, we applied EfficientNet_b0, which was already trained, as a feature extractor. In this phase, it is confirmed that the pre-trained EfficientNet_b0 can identify, generalize, and extract crucial chest X-ray features. In actuality, EfficientNets are a set of models that are derived from the EfficientNet_b0 base model. The experimental evaluation of the suggested architecture showed the EfficientNet_b0 in action. Two fully connected layers that are integrated with batch normalization, activation, and dropout, respectively, send the pre-trained EfficientNet_b0 output features to our proposed custom top layers. A visual representation of the EfficientNet_b0 design is shown in Fig. 5 (the basic building block of EfficientNet-B0).

All MBCConv blocks take the height, width, and channel of h , w , and c as input. C is the output channel of the two MBCConv blocks. (Note MBCConv, mobile inverted bottleneck convolution; DW Conv, depth-wise convolution; SE, squeeze-excitation; Conv, convolution.) Here, (A) and (C) are mobile inverted bottleneck convolution blocks, but the difference is that (C) is six times that of (A). (B) is squeeze-excitation block. A complete workflow of the MBCConv1, $k3 \times 3$ and MBCConv6, $k3 \times 3$ blocks is shown in Fig. 5. Both MBCConv1, $k3 \times 3$ and MBCConv6, $k3 \times 3$ use depth wise convolution, which integrates a kernel size of 3×3 with the stride size of s . In these two blocks, batch normalization, activation, and convolution with a kernel size of 1×1 are integrated. The skip connection and a dropout layer are also incorporated in MBCConv6, $k3 \times 3$, but this is not the case with MBCConv1, $k3 \times 3$. Furthermore, in the case of the extended feature map, MBCConv6, $k3 \times 3$ is six times that of MBCConv1, $k3 \times 3$, and the same is true for the reduction rate in the SE block, that is, for MBCConv1, $k3 \times 3$ and MBCConv6, $k3 \times 3$, r is fixed to 4 and 24, respectively. Note that MBCConv6, $k5 \times 5$ performs the identical operations as MBCConv6, $k3 \times 3$, but MBCConv6, $k5 \times 5$ applies a kernel size of 5×5 , while a kernel size of 3×3 is used by MBCConv6, $k3 \times 3$. Instead of random initialization of network weights, we instantiate ImageNet’s pre-trained weights in the EfficientNet model, thereby accelerating the training process. ImageNet has performed a great feat in the field of image analysis, since it is composed of more than 14 million images covering eclectic classes. The rationale for using pre-trained weights is that the imported model already has sufficient knowledge in the broader aspects of the image domain. As shown in several studies [5], there is reason for optimism in using pre-trained ImageNet weights in state-of-the-art CNN models even when the problem area (namely COVID-19 detection) is considerably distinct from the one in which the original weights were obtained. The optimization process will fine-tune the initial pre-training weights in the new training phase so that we can fit the pre-trained model to a specific problem domain, such as COVID-19 detection. For the feature extraction process of the proposed ECOVNet architecture applying pre-trained ImageNet weights is executed after the image augmentation process, as presented in Fig. 6.

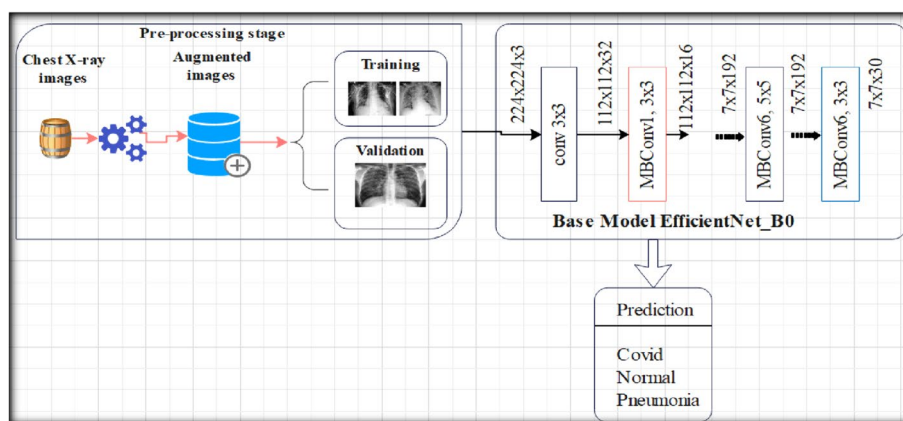


Fig. 6 An illustration of the suggested model architecture

Feature extraction using pre-trained Efficientnet_B0

Scaling up the fundamental network results in the creation of the EfficientNets family of models, which includes EfficientNet_B0 through B7 (generally known as EfficientNet_B0). EfficientNets have two benefits: first, it improves model performance by lowering parameters and FLOPS (floating-point operations that can be done by a computer unit in one second), and it boosts accuracy (Floating Point Operations Per Second). EfficientNets have attracted attention for their outstanding prediction performance by using a compound scaling strategy in the network's breadth, depth, and resolution. Mention how, on CNN, the terms "layers" and "resolution" refer to, respectively, the size of the image, while "width" and "depth" refer to the number of channels in each layer. Compound scaling, which asserts that as models get larger, the accuracy gain will decrease, is based on increasing the network's breadth, depth, or image resolution. The network dimensions are scaled consistently via compound scaling, which makes use of a compound coefficient. The compound coefficient scales the dimensions as shown in the graph below, depending on how many more resources are available for model scaling [4]. An EfficientNet can be defined by three dimensions: (i) depth, (ii) width; and (iii) resolution:

$$\begin{aligned} \text{Depth : } d &= \alpha^{\varnothing} \\ \text{Width : } w &= \beta^{\varnothing} \\ \text{Resolution : } r &= \gamma^{\varnothing} \\ \text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 &\approx 2 \\ \alpha \geq 1, \beta \geq 1, \gamma &\geq 1 \end{aligned}$$

where the possible scaling factors for each dimension that a grid search may set are α , β , and γ , and \varnothing is the compound coefficient. The target model size is obtained by scaling the baseline network (EfficientNet_B0) using the scaling factors that have been determined. For instance, when EfficientNet_B0 is used, $\varnothing = 1$ is specified, the optimal values, i.e., $\alpha = 1.2$, $\beta = 1.1$, and $\gamma = 1.15$, were gained using a grid search under the constraint of $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$. Scaling up EfficientNet_B0 to get EfficientNet_B1 to B7 requires adjusting the value of \varnothing in Eq. 1. Several mobile inverted bottleneck convolution (MBConv) blocks with integrated squeeze-and-excitation (SE), batch normalization, and Swish activation make up the EfficientNet_B0 baseline architecture's feature extraction [4].

The ensemble architecture of EfficientNet, or MBConv, has been shown to categorize images more accurately than conventional convolution while utilizing ten times less parameters and FLOPs. Table 2 displays specific details for each tier of the EfficientNet_B0 reference network. EfficientNet_B0 is composed of 16 MBConv blocks, each of which differs in terms of kernel size, feature map expansion phase, reduction ratio, etc. Figure 5 depicts the whole workflow of the MBConv1, $k3 \times 3$ and MBConv6, $k3 \times 3$ blocks. Both MBConv1, $k3 \times 3$ and MBConv6, $k3 \times 3$ employ depth-wise convolution, which combines a 3×3 kernel size with a stride size. Batch normalization, activation, and convolution with a 1×1 kernel size are merged in these two blocks. In MBConv6, $k3 \times 3$, a skip connection and a dropout layer are also included, but not in MBConv1, $k3 \times 3$. Additionally, MBConv6, $k3 \times 3$ is six times as fast as MBConv1, $k3 \times 3$ for the expanded feature map, and the SE block's reduction rate is fixed at 4 for MBConv1, $k3 \times 3$ and 24 for MBConv6, $k3 \times 3$,

Table 2 Detailed description of the EfficientNet_B0 base network layers

Stage	Operation	Resolution	#Output feature maps	#Layers
1	Conv 3 × 3	224 × 224	32	1
2	MBCConv1, k3 × 3	112 × 112	16	1
3	MBCConv6, k3 × 3	112 × 112	24	2
4	MBCConv6, k5 × 5	56 × 56	40	2
5	MBCConv6, k3 × 3	28 × 28	80	3
6	MBCConv6, k5 × 5	14 × 14	112	3
7	MBCConv6, k5 × 5	14 × 15	192	4
8	MBCConv6, k3 × 3	7 × 7	320	1
9	Conv 1 × 1 & Pooling & FC	7 × 7	1280	1

respectively. The procedures of MBCConv6, k5 × 5, which applies a kernel size of 5 × 5, are the same as those of MBCConv6, k3 × 3, which employs a kernel size of 3 × 3.

Label smoothing

Label smoothing is one of the many regularization techniques. Label smoothing will help you train your model with more robustness and improve performance even in the presence of noisy data. Label smoothing is pre-implemented in TensorFlow. You just need to pass the correct parameter value. It is used to address the issue of over confidence in a model since it is an uncalibrated model with consistently increased predicted probability. Label smoothing vector is a regularization approach where the loss function is cross-entropy and the model uses a softmax function to the pan ultimate logic layer to compute its probability. This is because it prevents the largest logic from being larger than the rest. Instead of using the one-hot encoded label vector hot, label smoothing combines y_{hot} and the uniform distribution:

$$y_{Is} = (1 - a) * y_{hot} + a/k \quad (1)$$

$$y_{Is} = \text{label smooth } y \text{ value}$$

$$k = \text{number of label classes}$$

$$y_{hot} = \text{one hot encoded label vector}$$

$$a = \text{hyper parameter that controls the degree of smoothing}$$

if $\alpha = 1$ we get uniform distribution and if $\alpha = 0$
we obtain the original one hot encoded y_{hot}

$$\text{Criterion} = nn. \text{CrossEntropyLoss}(\text{label}_{\text{smoothing}} = 0.11)$$

Classifier

The global averaged feature, which is then followed by a classifier, is the outcome of the EfficientNet design. We employed a two-layer MLP, also known as a fully connected (FC) layer, for the classification test. This MLP integrates elements of EfficientNet into two neural layers (each neural layer has 512 nodes).

We used batch normalization, activation, and dropout layers in between FC layers. Deep network training is significantly accelerated by batch normalization, which also increases neural network stability. The optimization process is smoother and the gradient behavior is more constant and predictable, which speeds up training. Sigmoid weighted linear units (SiLUs), whose activation is determined by multiplying the sigmoid function by its input, have been used in this work. When evaluated in 3 deep architectures on CIFAR-10/100, [8] discovered that the SiLU consistently outperforms the other activation functions:

$$f(x) = x \cdot \sigma(x) \quad (2)$$

The activation a_k of the k th SiLU for input z_k is obtained by multiplying the sigmoid function by its input.

$$a_k(z_k) = z_k \sigma(z_k). \quad (3)$$

$$\text{where sigmoid function is : } \sigma(x) = \frac{1}{1 + e^{-x}} \quad (4)$$

Instead of growing monotonically, the activation of SiLU has a global minimum value of around -0.28 for $z_k \approx -1.28$. The fact that SiLU has a self-stabilizing property is one of its key characteristics. The implicit regularizer that prevents the learning of weights with large magnitudes is the global minimum, which occurs when the derivative is zero. The weights are supported by it like a soft floor. We added a Dropout layer, one of the greatest regularization techniques, following the activation procedure to reduce overfitting and yield predictions that were more accurate. The capacity of this layer to arbitrarily delete particular FC layer nodes results in the removal of all arbitrarily selected nodes as well as all incoming and outgoing weights. Using either a validation set or a random estimate (i.e., $p=0.5$), p may be used to assess the likelihood of each layer's randomly selected nodes dropping. We kept the dropout rate at 0.2 throughout. The next optimizer utilized is AdamW, a variation of Adam with a better implementation of weight decay. The classification layer assigned the input chest X-ray images to the COVID-19, normal, and pneumonia classes using the softmax activation function to convert the activation from the previous FC layers into a class score.

Evaluation metric

In order to determine how well the given strategy performed, we used the evaluation metric:

$$\text{Accuracy} = \frac{TP + TN}{TN + FP + TP + FN} \quad (5)$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (6)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (7)$$

$$\text{F1 - score} = \frac{2 * TP}{2 * TP + FP + FN} \quad (8)$$

TP = True positive

TN = True negative

FN = False negative

FP = False positive

Results and discussion

In this section, the findings will be examined together with several experimental approaches, the suggested Efficientnet_b0's output, and the durability of the model. The Google Colab Notebooks platform is used to implement the suggested COVID-19 identification method in Python. Several cutting-edge techniques are used to assess how well the suggested model works in addressing the three-class categorization challenge. Determine if the chest X-ray image belongs in the COVID-19, normal, or pneumonia categories using the three-class categorization task. Scikit-Learn, PyTorch, and the TensorFlow back-end library make up the software stacks for all of our applications. In order to distinguish between X-ray images of Covid, normal, and pneumonia patients, the Efficientnet_b0 deep network model was trained. Weights are continually changing during deep network training, which entails backpropagating errors from the top layer to the bottom layer. During training, a number of model-related parameters were optimized. The network model's optimum parameter settings during training are (i) number of input last layer: 2048, (ii) batch size: 32, (iii) number of epochs: 150, (iv) optimizer: AdamW, (v) loss function: categorical cross-entropy (label smoothing) and (vi) activation function of the last (classification) layer: sigmoid weighted linear unit (SiLU). Trained Efficientnet_b0 model has 100% training accuracy. It is clear that the developed model predicts i) Covid with test accuracy of Covid: 96% (24/25), and (ii) normal with test accuracy of normal: 94.7% (18/19). (iii) Pneumonia patients with a test accuracy of 90% (18/20) for viral pneumonia. Total overall test accuracy is 93.75 (60/64.0). Out of 25 test images of Covid positive, 1 is misclassified as Covid positive. Out of 19 test images of normal, 1 is misclassified as normal images. There are 2 viral pneumonia images out of the 20 test images that were incorrectly identified as such. The overall classification accuracy of the designed model on the test dataset was 93.75%, which is a respectable level. Table 3 gives a detailed performance of the evaluation metrics used to

Table 3 Performance metrics of the Efficientnet_b0

Class	Accuracy (%)	Specificity (%)	Sensitivity (%)	F1-Score (%)
COVID-19	96	95.7	96.5	95
Pneumonia	90	91	87.9	89.3
Normal	94.7	94.6	94.8	94.6

assess the performance of the Efficientnet_b0 which give a good indication of the models performance.

Conclusion

The coronavirus pandemic has put a major strain on the healthcare systems of every nation on the planet due to the enormous number of deaths it has caused. Quicker, easier, and less expensive early illness diagnosis can help save lives. The COVID-19 has changed the problems that all people now encounter in their everyday lives. There are now a lot more people with this infection. Reverse transcriptase-polymerase chain reaction (RT-PCR), the standard technique of diagnosing the disease, has its own challenges in identifying true positives and true negatives, similar to immunodiagnostic testing of COVID-19-infected persons [20]. The test could also need a range of lab tools, depending on the ages of the individuals. This study uses a publicly accessible X-ray chest dataset from the Kaggle database to identify Covid-19 based on the characteristics obtained from the images. The images have been divided into three primary groups. Using efficientnet_b0 networks, we demonstrate a successful method for the automated detection of COVID-19. With chest X-ray images, this method splits the images into three groups. We classified the images using the three Covid classifications of normal, pneumonia, and Covid in the provided manner, and an average accuracy of above 90% was attained, which is quite encouraging when compared to the most current pneumonia detection techniques. Furthermore, CNN-LSTM [14] models, which have been frequently employed for the identification of pneumonia, are contrasted with the suggested Efficientnet_b0 model. The proposed model compared with Cornet proposed by [12] that has an accuracy of 89.60% and VGG16, VGG19, ResNet proposed by [17] also with an accuracy of 80%. The findings showed that the suggested model functioned faster and more accurately than the comparison models. Additionally, it is anticipated that employing this method will lower medical expenses, the likelihood that nurses or doctors may contract COVID-19 during swab collection, and future fatalities. Future research will expand the scope of chest X-ray images to include more illness classes and integrate them with other clinical data. Further, the size of the dataset will also be increased by looking at dataset from other databases.

Abbreviations

AI	Artificial intelligence
WHO	World Health Organization
RT-PCR	Reverse transcriptase-polymerase chain reaction
SiLU	Sigmoid weighted linear unit
CT	Computed tomography
CNN	Convolutional neural network
FC	Fully connected layer
TP	True positive
TN	True negative
FLOPS	Floating point operations per second
MBCConv	Mobile inverted bottleneck convolution
DWConv	Depth-wise convolution
SE	Squeeze-excitation
Conv	Convolution

Acknowledgements

We would want to thank everyone who on one way or the other assisted, as well as all of the researchers who helped us with the study; their research papers gave us a lot of inspiration.

Author contributions

MAJ obtained the dataset and served as a lead in building the model. PA served as a technical advisor and helped in the writing of the article. YMM assisted in build of the Efficientnet_b0 model. VAA contributed immensely in writing the manuscript and the model building.

Funding

No funding was received from any source.

Availability of data and materials

All dataset and codes are available from the corresponding author on reasonable request.

Declarations**Competing interests**

The authors declare that they have no competing interests.

Received: 27 November 2022 Accepted: 18 April 2024

Published online: 18 June 2024

References

1. Alazab M, Awajan A, Mesleh A, Abraham A, Jatana V, Alhyari S (2020) COVID-19 prediction and detection using deep learning. *Int J Comput Inf Syst Ind Manag Appl* 12:168–181
2. Badawy M, Ramadan N, Hefny HA (2023) Healthcare predictive analytics using machine learning and deep learning techniques: a survey. *J Electr Syst Inf Technol* 10(1):1. <https://doi.org/10.1186/s43067-023-00108-y>
3. Bassi PRAS, Attux R (2022) A deep convolutional neural network for COVID-19 detection using chest X-rays. *Res Biomed Eng* 38(1):139–148. <https://doi.org/10.1007/s42600-021-00132-9>
4. Chowdhury NK, Kabir MA, Rahman MM, Rezoana N (2020) ECOVNet: an ensemble of deep convolutional neural networks based on EfficientNet to detect COVID-19 from chest X-rays. <https://doi.org/10.7717/peerj-cs.551>
5. Chowdhury NK, Kabir MA, Rahman MM, Rezoana N (2021) ECOVNet: a highly effective ensemble based deep learning model for detecting COVID-19. *PeerJ Comput Sci* 7:1–25. <https://doi.org/10.7717/PEERJ-CS.551>
6. Coronavirus disease 2019 (COVID-19) – Images—BMJ Best Practice (nd) Retrieved October 21, 2022, from <https://bestpractice.bmj.com/topics/en-gb/3000201/images-and-videos>
7. Demir F (2021) DeepCoroNet: a deep LSTM approach for automated detection of COVID-19 cases from chest X-ray images. *Appl Soft Comput* 103:107160. <https://doi.org/10.1016/j.asoc.2021.107160>
8. Elfwing S, Uchibe E, Doya K (2018) Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw* 107:3–11. <https://doi.org/10.1016/j.neunet.2017.12.012>
9. Elpeltagy M, Sallam H (2021) Automatic prediction of COVID-19 from chest images using modified ResNet50. 26451–26463.
10. Fei-Fei L, Deng J, Li K (2010) ImageNet: constructing a large-scale image database. *J Vis* 9(8):1037–1037. <https://doi.org/10.1167/9.8.1037>
11. Islam MZ, Islam MM, Asraf A (2020) A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images. *Inf Med Unlocked* 20:100412. <https://doi.org/10.1016/j.imu.2020.100412>
12. Khan AI, Shah JL, Bhat MM (2020) CoroNet: a deep neural network for detection and diagnosis of COVID-19 from chest x-ray images. *Comput Methods Programs Biomed* 196:105581. <https://doi.org/10.1016/j.cmpb.2020.105581>
13. Minaee S, Kafieh R, Sonka M, Yazdani S, Jamalipour Soufi G (2020) Deep-COVID: predicting COVID-19 from chest X-ray images using deep transfer learning. *Med Image Anal* 65:1. <https://doi.org/10.1016/j.media.2020.101794>
14. Mousavi Z, Shahini N, Sheykhiand S, Mojtahedi S, Arshadi A (2022) COVID-19 detection using chest X-ray images based on a developed deep neural network. *SLAS Technol* 27(1):63–75. <https://doi.org/10.1016/j.slast.2021.10.011>
15. Narin A, Kaya C, Pamuk Z (2021) Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. *Pattern Anal Appl* 24(3):1207–1220. <https://doi.org/10.1007/s10044-021-00984-y>
16. Roosa K, Lee Y, Luo R, Kirpich A, Rothenberg R, Hyman JM, Yan P, Chowell G (2020) Real-time forecasts of the COVID-19 epidemic in China from February 5th to February 24th, 2020. *Infect Dis Model* 5:256–263. <https://doi.org/10.1016/j.idm.2020.02.002>
17. Sahinbas K, Catak FO (2021) Transfer learning-based convolutional neural network for COVID-19 detection with X-ray images. In: *Data science for COVID-19 volume 1: computational perspectives*. Elsevier Inc. <https://doi.org/10.1016/B978-0-12-824536-1.00003-4>
18. Silva P, Luz E, Silva G, Moreira G, Silva R, Lucio D, Menotti D (2020) COVID-19 detection in CT images with deep learning: a voting-based scheme and cross-datasets analysis. *Inf Med Unlocked* 20:100427. <https://doi.org/10.1016/j.imu.2020.100427>
19. Tembhurne J (2024) Classification of COVID-19 patients from HRCT score prediction in CT images using transfer learning approach. *J Electr Syst Inf Technol* 11(1):1. <https://doi.org/10.1186/s43067-023-00129-7>
20. Teymouri M, Mollazadeh S, Mortazavi H, Ghale-noie ZN (2021) Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information. *Pathol - Res Pract* 221:153443
21. Zhong Z, Zheng L, Kang G, Li S, Yang Y (2017) 2017(cites:1625)Random Erasing Data Augmentation

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.