Received: 28 February 2023 | Revised: 11 July 2023 | Accepted: 3 August 2023 | Published online: 13 September 2023

RESEARCH ARTICLE

Artificial Intelligence and Applications
2025, Vol. 3(1) 94–109
DOI: 10.47852/bonviewAIA3202819

BON VIEW PUBLISHING

COVID-19 Mortality Risk Prediction Using Small Dataset of Chest X-ray Images

Akeem Olowolayemo[1,*], Wafaa Khazaal Shams[2], Abubakar Yagoub Ibrahim Omer[1], Yasin Mohammed[1] and Raashid Salih Batha[1]

[1]*Department of Computer Science, International Islamic University Malaysia, Malaysia*

[2]*Ministry of Higher Education and Research, Iraq*

Abstract: COVID-19 outbreak ravaged the whole world starting from the early part of 2020. The rapid spread of the pandemic accounts for the major reason the world was thrown into panic mode and pervasive confusion. However, COVID-19's greatest strength is its virility, but its severity on an individual is mostly ambiguous, which is dependent on the particular individual. This, combined with the increasingly limited capacity of the global healthcare infrastructure, warrants some mechanism that can predict the prognosis of an individual to better determine if the patient would require hospital resources or be better treated as an outpatient. The lack of such a mechanism leads to suboptimal utilization of valuable hospital resources leading to unnecessary loss of life. However, often at the onset of a pandemic such as it was experienced during the outbreak of COVID-19, ample and appropriately labeled dataset to build accurate deep learning models to assist in this respect was limited. In this vein, frantic efforts were made to acquire dataset to train deep learning models for the stated objectives, unfortunately only a small dataset from a single source was available at the time of the study. Consequently, deep learning models based on the ResNet-18 architecture were trained on a small dataset of chest X-rays of patients infected with COVID-19 to predict mortality risk. The models exhibit considerable accuracy with high sensitivity. The appropriateness of the techniques proposed in this study for predictive modeling may be particularly suited when only small datasets are available especially at the onset of similar pandemics. From existing literature, models with low complexity such as ResNet perform better with small dataset. Hence, this study utilized ResNet-18 as the baseline to evaluate the performance of other popular models on small datasets. The performance of the baseline models based on ResNet-18 with an accuracy of 0.89 compared favorably with those of the several other models including AlexNet, MobileNetV3, EfficientNetV2, SwinTransformer, and ConvNeXt using the same datasets and similar parameters.

Keywords: small dataset, deep learning, convolutional neural networks (CNNs), X-rays image classification, COVID-19 mortality

1. Introduction

Originating in late 2019, COVID-19 has since ravaged the globe. The spread is unprecedented due to globalization. The combination of its fast spread and potency of its infections make it an especially effective disease, however, with a relatively low mortality rate. COVID-19's virility in particular has tested the limits of the global healthcare infrastructure due to the sheer number of people being infected in a short span, putting the global healthcare systems under intense pressure. The consequences of this can be seen best in China where a stadium had to be transformed into a specialty hospital [1], or more recently in the case of India where the army set up temporary hospitals in various parts of the country [2].

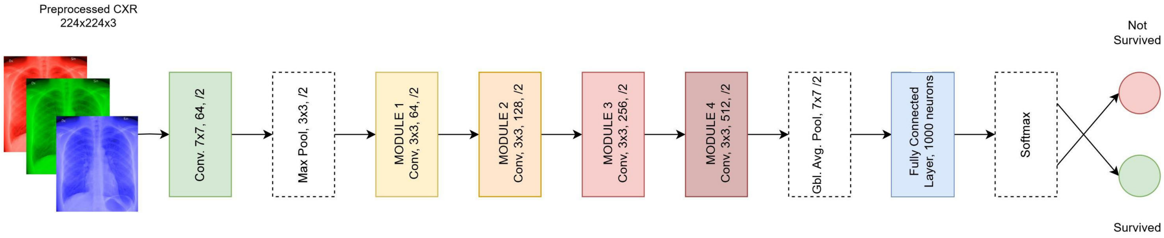
A significant aspect to consider is the variation in severity for the infected patients where symptoms can range from mild to severe. This means that not everyone requires hospitalization and while certain co-morbidities factors, such as age and diabetes, for instance, have been isolated as high risk, other factors have not been clearly identified as been predisposed to severity or mortality. Therefore, it is not known for certain whether a particular individual needs hospitalization, or if they can just recuperate at home. If this could be determined early on, limited healthcare resources can be prioritized effectively for those classified as predisposed to mortality. This is the most relevant aim of this study.

These two facets of the pandemic, specifically limited capacity of the healthcare system and the undetermined need for hospitalization, have presented the need for some prioritization mechanism to determine which patients require hospitalization. However, due to the aforementioned ambiguity in severity, there is a lack of such mechanism, which has led to a rather morbid result. Certain patients are hospitalized, utilizing valuable resources when they do not need it, while others have no options, dying due to the lack of resources. This has subsequently forced the hands of doctors to make rather poignant yet necessary decisions when it comes to who gets to use such resources.

*Corresponding author: Akeem Olowolayemo, Department of Computer Science, International Islamic University Malaysia, Malaysia. Email: akeem@iium.edu.my

Dr. Wafaa Khazaal Shams has changed the affiliation to Ministry of Higher Education and Research, Iraq.

Figure 1
ResNet-18 architecture



From the foregoing, it is believed that a potential solution can be found through the classification of chest X-rays (or CXRs) of the patients early on, to predict the need for hospitalization based on potential mortality risk. Radiologists commonly use CXRs to determine lung abnormalities in both emergency and non-emergency scenarios through visual analysis. This is no different when it comes to COVID-19 where pneumonia is the most common manifestation. CXRs have been shown to detect such lung abnormalities with adequate sensitivity and specificity [3].

This problem of suboptimal hospitalization has presented the domain of deep learning an opportunity to stretch its wings. Deep learning models can detect the underlying patterns in images that may not be perceptible to the human eye. A model that can predict the probable mortality risk of a COVID-19 patient utilizing CXRs may be an invaluable asset to the healthcare providers globally by serving as an appropriate prioritization mechanism. Consequently, it may assist to prevent unnecessary loss of life by optimal allocation of medical resources.

The uncertain need for hospitalization of patients infected with COVID-19, combined with the increasingly limited capacity of the healthcare system, can lead to suboptimal hospitalization, which deprives deserving patients of emergency care, resulting in unnecessary loss of life. Consequently, this research aims to understand how deep learning and computer vision techniques can be employed specifically in the domain of X-rays analysis for COVID-19 patients' potential mortality classification. The only reliable dataset on COVID-19-related X-ray is utilized for the source of. The data utilized for the research were obtained from and were subsequently transformed into a viable format appropriate for the deep learning model developed for the classification approach, to train and tune the deep learning model for the prediction of mortality risk. The performance of the resulting deep learning models was evaluated to understand its strengths and weaknesses to further improve its solution efficacy. This is crucial in order to achieve an optimal predictive model.

It is hoped that with a prospective mortality classification model based on deep learning, unnecessary loss of life can be prevented by optimally prioritizing for patients in the face of medical resource scarcities, as well as lowering the number of COVID-19 patients' hospitalization. Consequently this lead to prevent further exposure by decreasing number of people in hospitals, reduces peak infection incidence, protection of the well-being of the medical staff, and delays occurrence of the said peak as well as reduces the cumulative number of infections during the pandemic.

Since COVID-19 outbreak and its impacts were unprecedented, acquiring appropriate dataset to build predictive modeling was extremely challenging at the onset of the COVID-19. Medical and health practitioners were more focused on combating the onslaught of the pandemic than structuring useful data coupled with the fact that a global structure for handling datasets for studies was not sufficient in place. In our frantic search, we eventually got an extremely mingre dataset collected for several purposes but which was possible to extract the CXR images and other features that were suited for the intended predictive models. Consequently, techniques that may possible better on small datasets were the main focus of our search for the predictive solutions. From previous studies, models with low complexities have been found to performance better in situations where only small datasets were available. In Brigato and Iocchi [4], ResNet models, a set of decent models, have been shown to be suitable for predictive modeling where a small dataset is available. When there are only minimal datasets available, such as during the start of similar pandemics, the methodologies for predictive modeling suggested in this study may be especially well suited. According to the research currently available, models with modest complexity, like ResNet-18, perform better with small datasets. Hence, in this study, ResNet-18 has been used as a baseline to compare the effectiveness of other widely used models on tiny datasets.

The rest of this paper is organized as follows. The next section highlights the related work in the areas closely related to the prediction of mortality to guide the allocation of healthcare resources, this is followed by the methodology section, which described the formulation of the algorithm, preparation of the data, and overall strategy to ensure considerable performance. The subsequent section is focused on performance and discussion of results, while the conclusion section wrapped up the study.

2. Related Work

Previous studies such as Wang et al. [5] acknowledge the high infection rate of COVID-19 and the undesirable effects that it causes with regard to restricted healthcare facilities and avoidable casualties. This study is motivated primarily to provide quick diagnosis for patients with COVID-19 symptoms that show worse prognosis for early prevention before the onset of severe symptoms. In that vein, the study proposes an automatic convolutional neural networks (CNNs) model for COVID-19 diagnostic and analysis of prognostic by routinely utilized computed tomography (or CT). The choice of CT due to the fact that in contrasts to RT-PCR tests, CT is much more sensitive, even for patients who are asymptomatic and can be acquired quickly without additional costs involved. The model demonstrated considerable performance, able to distinguish COVID-19 from other pneumonia with area under the receiver operating characteristic (ROC) curve (AUC) of 0.87 and 0.88, respectively, and viral pneumonia with AUC 0.86. More importantly, the deep learning system succeeded in stratifying patients into high-risk and low-risk groups based on hospital stay duration.

Kulkarni et al. [6], on the other hand, chose an opposing view compared to Wang et al. [5] opted not to use CT scans but rather CXRs. However, similar to the earlier-mentioned paper, the study acknowledged that mechanical ventilators were scarce and there is a crucial requirement to utilize them optimally by adequately rationing them. Consequently, this paper proposes a deep learning model to predict the need for mechanical ventilation using CXR images in hospitalized patients with COVID-19. They argued that it was more practical to utilize X-rays rather than CT scans due to being more widely used, easily available, and less likely to be affected by machine contamination. The performance of the model was considerable. The accuracy was 90.06%, and sensitivity was 86.34%, while specificity was 84.38%. The performance results were compared with the evaluation of two respiratory and intensive care specialists. This is to decide whether there is a need for mechanical ventilation or not. The model was found to outperform the specialists' predictions with an increased accuracy between 7.24% and 13.25%.

Luz et al. [7], like Santa Cruz et al. [8], highlight a potential hitch when it comes to models trained on X-ray with regard to COVID-19: accessibility and availability. The paper showed that many models have been developed with encouraging results but are not computationally prudent and efficient. In response to this problem, the paper proposes a method to provide a more efficient and effective model for identification of COVID-19 using CXRs. This is done via a set of EfficientNet CNN models combined with a hierarchical classifier. The research demonstrated encouraging performance, with an accuracy of 93.9% and parameters 5–30 times fewer compared to several existing architectures that were tested.

Another related study was conducted by Santa Cruz et al. [8], however as an exploratory study. In the study, the focus was on an important issue when it comes to models built to aid medical practitioners in clinical use, specifically bias induced by lack of quality control and lack of bias assessment of public COVID-19 datasets. The study acknowledges the existence of several machine learning models proposed to aid in the diagnosis and prognosis of COVID-19. It equally recognizes the short period in which this was done and the inherent availability of unnoticed limitations due to bias that inhibits the models from performing well on unseen or new testsets. The paper cautions on the inappropriate or misuse of these models and directs to new appropriate datasets that were becoming available to researchers. It also highlights best practices for modelers when it comes to choosing datasets.

In Cheng et al. [9], the authors used longitudinal CXR with clinical data to predicate the mortality of COVID-19 patient. By applying longitudinal transformer-based network, the accuracy was 0.732. Another study shows the effect of using logistic regression model with CXR and clinical variables to predict the hospital length of stay achieving an AUC of 0.87 while prediction accuracy is 0.78 for those requiring the use of oxygen supplementation [10]. Combination between chest radiograph images with clinical variables improves the predication of severe COVID-19. This is also indicated in the study by Munera et al. [11], using training CNNs for chest radiograph images and clinical variables that identified by random forest method. The results were 0.92 accuracy for patient admission and 0.81 for hospital mortality. This study was done on 2552 patients.

Previous study done by Islam et al. [12], had applied a deep learning based approach using Densenet-121 to effectively detect COVID-19 patients. Also the (CheXNet)model was utilized to weight the information regarding radiology image. The model was trained and tested on COVIDx dataset containing 13,800 chest radiography images of a total of 13,725 patients. By considering both two-class and three-class classifications ,that was able to achieved an accuracy of 96.49% and 93.71% respectively. Wang et al. [13], proposed Vision Transformer (VIT)–based model called PneuNet. The multi-head attention was applied on channel patches rather than feature patches to overcome the difficulty that lies in accurately identifying and classifying pathological features of COVID-19. Patients with mild symptoms do not show a marked difference in lung texture compared to those COVID-19 negative. Consequently, traditional CNNs do not perform considerably well in the classification of CXR images for the diagnosis of pneumonia. The proposed PneuNet was applied on a combined CXR dataset and reached 95.13%accuracy.

The study by Brigato and Iocchi [4] demonstrated that given small data, employing low complexity models in some setups can advance the state of the art. For instance, it showed that low-complexity CNNs outperform state-of-the-art architectures on problems with few training samples and without data augmentation. This study consequently adopted ResNet-18 as the baseline to evaluate the performance of the rest of the popular models in providing suitable model for mortality prediction.

This study presents an attempt to determine COVID-19 patients' mortality risk utilizing the available small dataset of CXR images at the onset of the pandemic. The study is completely different from the aforementioned studies as well as other existing studies on application of image classification to COVID-19 X-ray images [13–17] in two folds. Firstly, it focuses specifically on mortality risk prediction rather than COVID-19 positive identification in patients or the general disease identification. Secondly, most groundbreaking existing studies were often trained on massive datasets; this study focuses on utilizing small dataset to achieve predictive modeling in a critical period at the onset of COVID-19 pandemic.

3. Materials and Methods

Altogether, the methodology for this study consists of cleaning of the dataset, imputation of the missing values, augmentation of the images to achieve a sizeable dataset, creation of the data pipeline, transformation of the images, training and generation of the models, as well as evaluation. CNNs are set of models specifically utilized for the classification of image datasets, identification and recognition of still or moving objects, segmentation of objects in images, as well as general computer vision applications. Therefore, a model that is based on CNNs called residual networks (or ResNets) [11] is utilized for this study shown in Figure 1. ResNet-18 is selected as the baseline model and architecture of choice due to its appropriateness and performance on small dataset. The architecture is as shown in the figure below. The number 18 indicates the number of layers that the architecture possesses. ResNet-18 was chosen because it has the benefit of rapid convergence. The steps in the ResNet-18 models are discussed in the next subsections.

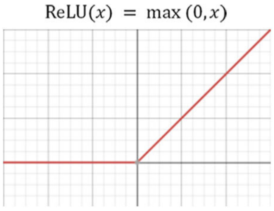
3.1. Convolutional layer

ResNet-18 CNNs as the baseline algorithms were applied on COVID-19 CXRs images present in the only available dataset found at the early stage of the pandemic [13]. The performance of the models was then compared with that of several popular CNN models discussed in the results and concluding sections. The processes to make the dataset appropriate for classification by ResNet-18 models are discussed under dataset subsection. The preprocessed COVID-19 CXR images initially are passed through a 7 × 7, 64-channel convolutional layer, responsible for feature extraction. Then, a process of convolution operations is carried out with a mask (or a kernel or a filter) of a predetermined size moved across the images of the CXRs resulting in a feature map. The process of the convolution of the mask across the images is defined as:

$$G[i,j] = (C_{XR} * K)(i,\ j) \ = \ \sum_{u=-k}^{k}\sum_{v=-k}^{k} K[u,\ \mathrm{v}] * C_{XR}(i+\ u,\ j+v) \tag{1}$$

where K is the kernel or mask or filter in convolution operation across the X-ray images, C_{XR}.

Figure 2
The ReLU graph



The process comprises multiple convolutional layers stacked together in a series, which can be observed in the modules, where the output from one layer is fed as the input into the successive layer. The purpose of this process is to extract higher-level features. The hyper-parameters considered include the kernel dimensions and stride. Stride is the total steps required for the kernels. In this case, the kernel dimensions are 7 × 7 and operate on a stride value of 2.

The activation function chosen is the rectified linear activation function (ReLU). Images are intrinsically non-linear hence, the need for ReLU to transform into non-linearity at every convolutional layer. This improves model performance, and the model also converges quicker in comparison to common non-linear functions [12]. The ReLU is as seen in Figure 2 subsequently:

$$\mathrm{ReLU}(x) \ = \ \max(0, x) \tag{2}$$

3.2. Pooling layer

After the initial convolution, the resultant feature map tensor of the CXR encounters a 3 × 3 max pooling operation. The important function of the pooling layer is subsampling, to reduce the dimensions of the feature maps obtained from the convolutional layer. This is necessary to decrease the parameters of the network, and consequently, the computation required. Common pooling operations include max, average, and min pooling for extracting maximum, average, and minimum values, respectively, from the image segments matching the kernel or filter, at every stage. In similar fashion to the convolutional layer, the pooling layer equally considers parameters such as the layer dimensions and the stride. In the first max-pooling operation, a dimension of 3 × 3 and a stride of 2 were chosen.

3.3. Residual connections

The layers in the ResNet-18 model display unconventionality when compared to traditional CNN architecture as residual connections. This is due to poorer performance of deeper plain networks vanishing or exploding gradients. Residual connections are used to learn functions that reference the input, rather than the alternative which do not reference the input. These functions are called residual functions.

Figure 3
Residual building block

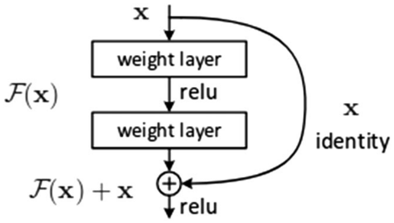

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 3 [18] depicts a residual block. $H(x)$ represents the anticipated output from the block, and x is the input value. The aforementioned residual function is defined in terms of x as

$$F(x) \ = \ H(x) - x \tag{3}$$

Due to this, the desired output is redefined as

$$H(x) \ = \ F(x) \ + \ x \tag{4}$$

The operation which results in $F(x) \ + \ x$ (which is equal to (x)) at the output is performed via the residual connection acting as a shortcut connection and element-wise addition. In this particular example consisting of two layers, the residual function is obtained by

$$F(x,\ \{W_i\}) = W_2\,\sigma(W_1 x) \tag{5}$$

where denotes the activation function, ReLU.

The intuition for redefining the output $H(x)$ with respect to x is to make it easier for the multiple non-linear layers to optimize the referenced residual mapping $(H(x) - x)$ than the unreferenced $H(x)$ directly. In rare situations when the optimal output results in an identity mapping, that is, the output equals the input, x, the residual $F(x)$ naturally moves to 0. This is much easier than fitting the non-linear layers directly into the identity mapping.

There are two types of residual connections:

1) The identity shortcut is utilized when the dimensions of the input and output are the same. It is represented by

Figure 4
PA view and AP view

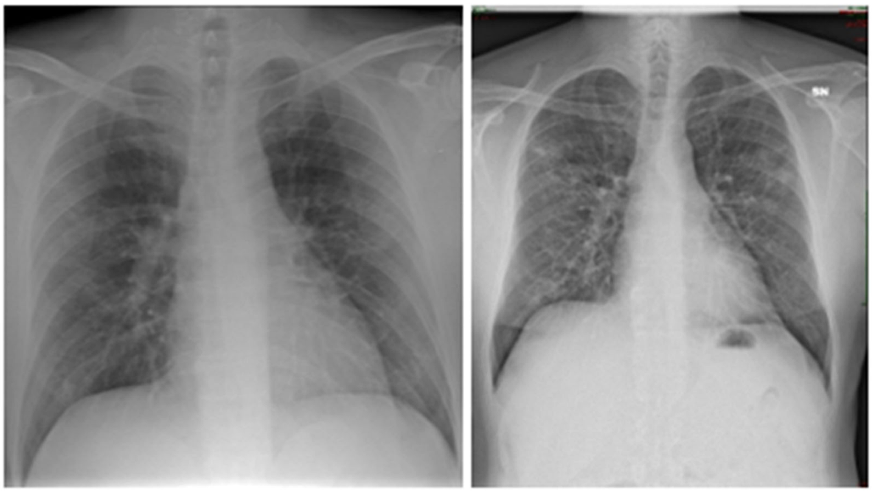


Figure 5
Sample demonstrating X-ray variety

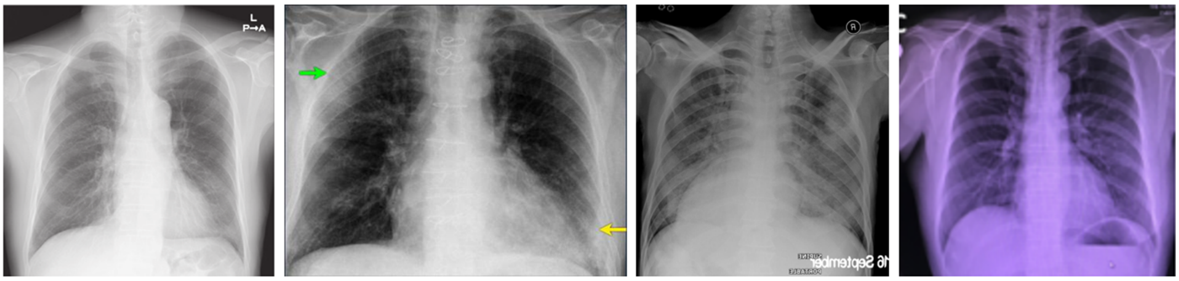
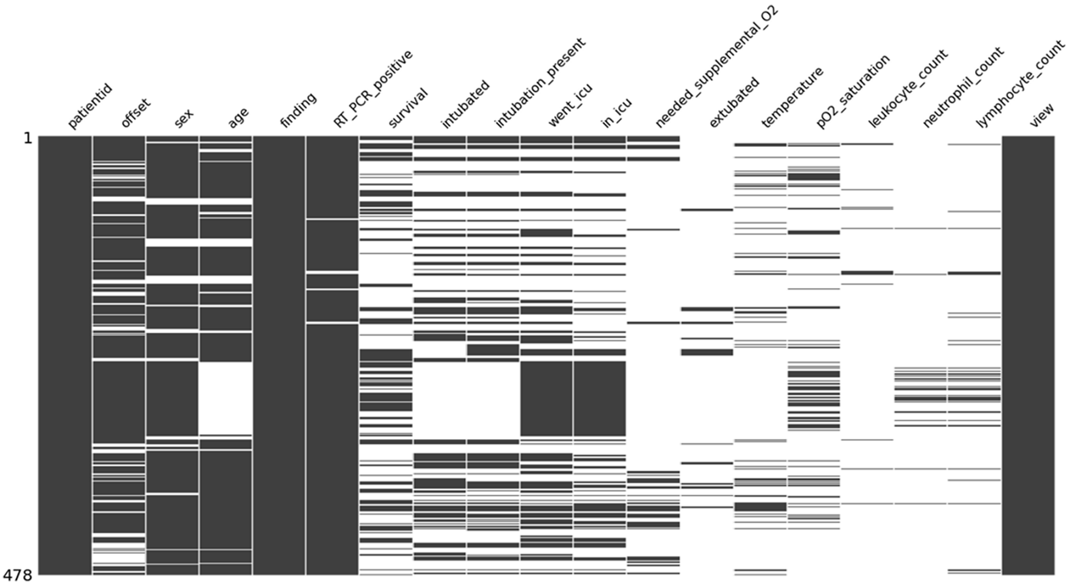


Figure 6
Missing values visualization



$$y = F(x, \{Wi\}) + x \tag{6}$$

2) The projection shortcut is utilized in situation where the dimensions of both the input and the output are different as a result of change in the number of channels between layers. In this case, a linear transformation is employed to ensure dimension parity. This has the side effect of introducing more parameters, represented by *Ws*.

$$y = F(x, \{Wi\}) + Ws\,x \tag{7}$$

3.4. Modules 1-4

After max pooling, the CXR tensor passes through four distinct modules connected in series. Generally, CNNs consist of repeating modules comprising the convolutional layer, the ReLU, and the pooling layers before reaching the fully connected layer. Each module consists of four convolutional layers of dimensions 3 × 3 and the same number of channels, ReLU activation functions, and accompanying residual connections. The channels of the convolutional layers of each module increase with 64 channels in the first module, 128 channels in the second module, 256 channels in the third module, and 512 channels in the fourth module. Since the number of channels changes between modules, the projection shortcut mentioned in Equation 7 is utilized. All the layers mentioned operate on a stride value of 2.

3.5. Fully connected layer

The CXR feature map tensor encounters the global average pooling operation after passing through the last module before it reaches the fully connected layer consisting of 1000 neurons. The fully connected layer appears at the end of the network and is responsible for the actual classification. It is representative of the traditional neural network architecture where all the neurons of one layer are connected to all the neurons of the next, forming a dense network. The tensor of feature maps is obtained from the layers before it is converted into a vector by a process referred to as flattening since the fully connected layers cannot operate on the CXR tensors directly. This is also why the layer is known as the flattening layer. Utilizing the features extracted from the CXR, the fully connected layer classifies the patient as potentially surviving or not surviving. Before the classification occurs, the softmax operation needs to be performed. It ensures that the output class probabilities add up to 1 and is defined as

$$\mathrm{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{8}$$

Each neuron's output in the layer with fully connected neurons is derived by

$$f(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{b}) = \boldsymbol{x}^T \boldsymbol{w} + b \tag{9}$$

This describes artificial neural networks in general. Here, *x* is the tensor of feature maps obtained from the CXR, *w* stands for the weight associated with the specific neuron, and *b* is the bias value.

3.6. Dataset

This study is focused on determining a COVID-19 patient's mortality risk based on a CXR. At the early part of the onset of COVID-19, it was challenging to locate a public COVID-19 CXR dataset with labels indicating mortality. Only one public dataset was found to have labels. Later dataset [19] was utilized in the extension of this study on large dataset [20]. The required dataset was collected from the initial small dataset found from a GitHub repository of radiography images of pneumonia patients made available in Wang et al. [13] consisting of 950 radiography images, out of which 584 are of X-rays and CT scans of COVID-19 patients, respectively. This repository not only had X-rays but also CT scans of COVID-19 patients along with patients that are infected with other viral pathogens such as SARS, MERS, influenza, herpes, etc. Since the focus of this project was only on X-rays of COVID-19 patients, the data were filtered to obtain a subset that contained X-rays for only those patients that are infected with COVID-19. Similarly, the dataset also contained X-rays in different views, namely anteroposterior (AP) view, posteroanterior (PA) view, and lateral (L) view. Only X-rays in the AP and PA views were extracted for this study. The AP and PA views are shown in Figure 4. After carrying out the above filtration, the number of images extracted from the dataset dropped to 478 out of the 950 images.

However, even most of the X-rays in the dataset extracted at this point had the survival labels missing. The remaining dataset was found to be very imbalanced, with a disproportionately high number of people that survived COVID-19 as compared to those that did not survive as shown in Figure 5. Therefore, two-class balancing techniques were applied and tested, namely class weight adjustment and weighted random resampling. To verify the efficacy of the imputed data along with the proposed class balancing techniques, three sets of different models were trained and evaluated for both the original and unimputed data, consisting of a model with no class balancing, a model with class weight adjustment, and a model with weighted random resampling.

3.6.1. Dataset and CXRS

The images in the dataset have been gathered from several different sources such as research publications and online radiography image databases. Consequently, the images are not always of the same dimensions or quality. Some X-ray images also contain annotations while others have a slight tint of blue or red.

This variety in data becomes useful as it enables the trained model to generalize well. However, the size of the dataset also is a crucial determining factor when it comes to generalizability. Therefore, although the variety of X-rays helps generalizability, the number of images and hence variations in the dataset are limited.

The first important consideration in preprocessing images used to train a CNNs, is that must all be the same size. For this study, all of the X-rays were cropped and resized to 224 × 224 because that resolution was the standard for image classification. The images were additionally normalized in similar fashion to how numerical data are normalized. The goal is to place 0 in the middle of a range of data. Tensors are used to depict images, typically representing three-dimensional arrays of numbers for the three-color channels (RGB) for each color image pixel. Subsequently, image normalization is carried for each of the channels, resulting in three separate normalizations.

The usual approach is to normalize images using ImageNet's mean and standard deviation. ImageNet is a database of millions of images used to train pre-trained models like ResNet-18.

In order to increase the number of image samples to avoid overfitting and possibly improve results, the training set's images were further augmented. Augmentation is accomplished by employing image transformation to change specific aspects of the current images and then training with the newly created images. In this instance, zoom, warp, brightness correction or adjustment, and rotation are among the image modifications that were attempted.

The attribute "survival" served as the primary focus of the data because the article's goal was to determine mortality risk. The "mortality" column, which must have values that are the opposite of those in the "survival" column, would be derived using this feature. However, the "survival" column had many of its values missing which made the X-rays associated with them unusable and also decreased the amount of data that were available for training to 169.

3.7. Imputing missing labels using a random forest classifier

One of the ways of overcoming the problem of missing values, showing in Figure 6, is through imputation. According to Aslani and Jacob [14], random forest-based imputation is the best method to accurately impute missing values in a column. Consequently, a random forest classifier (RFC) model is trained on the other features of the dataset to predict the missing values for the survival column, set as the target variable. The trained model can then be used to impute the missing values in the column and hence resolve the issue. Features for the RFC model to predict the "survival" column include information such as the patient's age, intubation status, and whether the patient is in ICU or had been to the ICU. ICU history (went_icu), oxygen assistance (need_supplemental_O2), and intubation status (intubation_present, extubated) were found to be good predictors of the "survival" label and hence used as features for the RFC model. The continuous variable age was binned into specific age groups so that it acts as a categorical variable instead. Moreover, it is well-known that the mortality risk from COVID-19 is different for different age groups, and hence it made sense to use age categories for predicting the survival label instead of a continuous value. The categories for the ages are: 0–30, 31–40, 41–50, 51–60, 61–70, 71–80, and 81–100. Therefore, to ensure that the performance metrics of the trained model were accurate, k-fold cross-validation was used. The number of splits for the k-fold cross-validation was chosen to be 5. After the RFC model was trained with the labeled data (accuracy ~92.5%), it was then used to impute the missing survival labels in the metadata.

The accuracy reached by the RFC model was only due to the highly indicative features, namely features that describe the patient's ICU history, oxygen assistance, and intubation status. These features already indicate that the patient is in a critical state and hence it is not surprising that these can be used to determine the survivability of the patient very well. However, since these features are only available after the patient enters the critical stage, such a model is not viable to be used for prognosis, unlike our model which just uses X-rays that are available before the patient reaches the critical state. Therefore, the high accuracy of the RFC model does not conflict with our methodology of using CXRs to predict mortality instead as the latter can be used for prognosis which is one of the objectives of our paper.

3.8. Class balancing

Two techniques were employed for class balancing and evaluated against the baseline for efficacy, viz: class weight adjustment and weighted random resampling.

3.8.1. Class weight adjustment

In this study, the weight for each instance of a class is changed to balance the total weight of either class. In the case of binary classification, the weights to be assigned to each class are found by getting the reciprocals of the number of positive and negative cases in the training set. The lower weight is assigned to the majority class and the higher weight is assigned to the majority class by passing them as an argument to the loss function. However, different scaling of weights of the classes can sometimes affect the behavior of certain loss functions. Since the reciprocal of the number of instances is being used, the resultant number may be very small in one case and very large in the other. To avoid this, the weights may be multiplied by half of the total number of cases in the training set. This normalizes the weights so that the weight scale is more similar in magnitude to the original values.

3.8.2. Weighted random resampling

In weighted random resampling, weights of each class in the training set are calculated in the same way as the class weight adjustment method, but instead of being assigned to a loss function, the weights are then used to create a new weighted sampler. This sampler is used during training to fetch samples from the training set, and based on the weights given for each class, the sampler would try to pass the same number of samples for each class in a batch, every time a new batch of samples is required during training. This means that in many cases, there may be multiple instances of the same image in the batch, usually of the minority class. The sampler does this to balance the number of instances from each class in the batch when there is a lack of sufficient unique samples from one or both classes. However, the images may not be exactly identical, as image augmentation is also applied to the images in the batch to create more variations in the data.

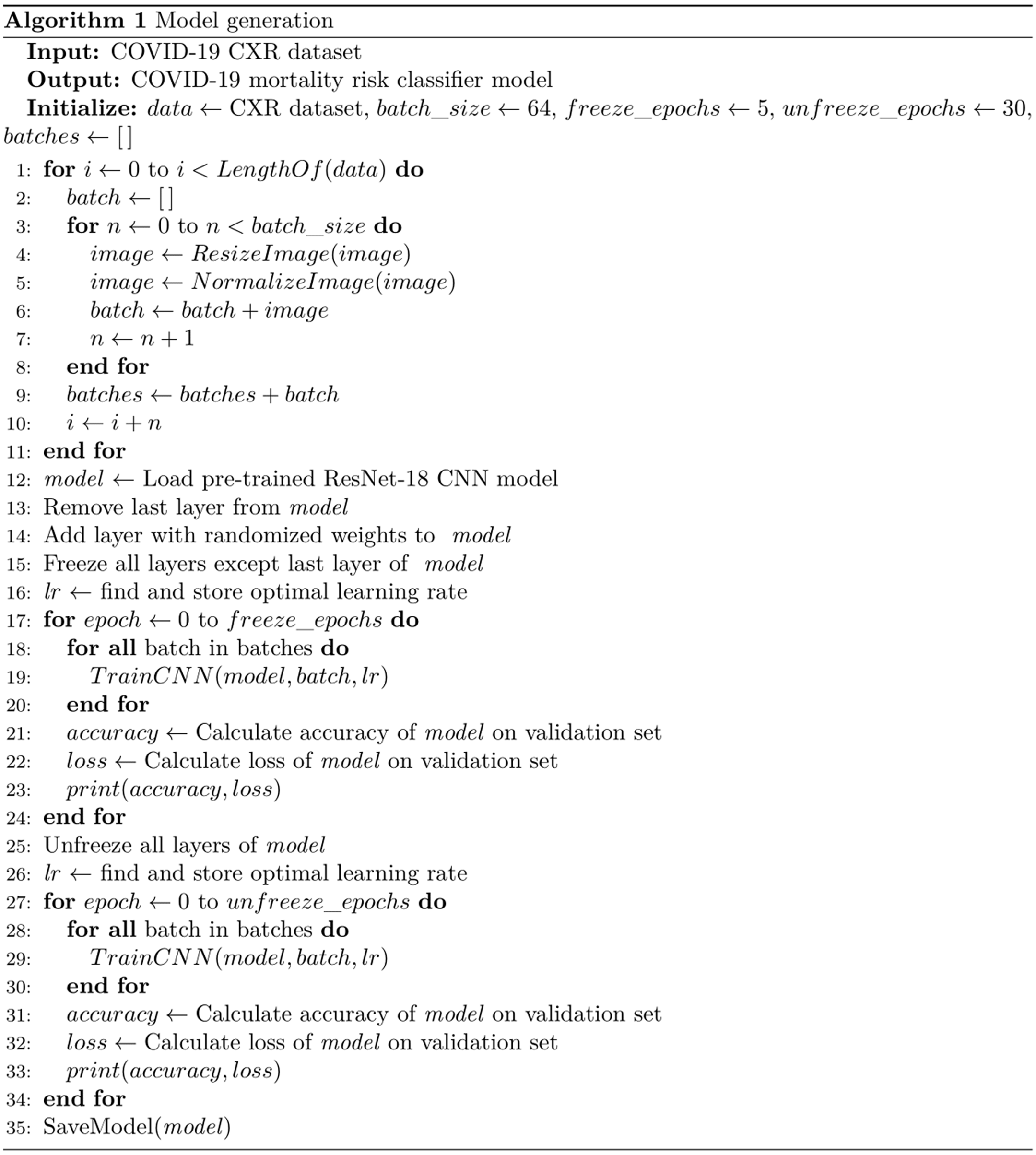
3.9. Preparing the data pipeline and applying image transformations

The images used to train a CNN must all be the same size. In the case of this paper, all of the X-rays were cropped and resized to 224 × 224 because that resolution was the industry standard for image classification. Furthermore, to train a CNNs model with a single image at a time is considerably time-consuming. In the alternate, training can be executed using a batch of images at each epoch, utilizing GPUs to quicken the processing. The DataBlock API of FastAI can be utilized to resize, normalize, enhance, and batch the images.

3.10. Model generation and evaluation

To demonstrate the performance improvements of our chosen techniques of imputation and class balancing, three models were trained on the original data and another three models were trained on the imputed data, consisting of a model with no class balancing, a model with class weight adjustment, and a model with weighted random resampling. Additionally, a stratified k-fold cross-validation method with a 5-fold split was employed for the training. In other words, the data were divided and evaluated five times, while maintaining the ratio of the target classes in each split, and each time, a different split of the data was utilized as the validation set, consequently training and evaluating the model on all dataset at the end of the five iterations. Due to the limited quantity of training data in this study, testing the methods on the

Figure 7
Model generation algorithm



```
Algorithm 1 Model generation
  Input: COVID-19 CXR dataset
  Output: COVID-19 mortality risk classifier model
  Initialize: $data \leftarrow$ CXR dataset, $batch\_size \leftarrow 64$, $freeze\_epochs \leftarrow 5$, $unfreeze\_epochs \leftarrow 30$, $batches \leftarrow [\,]$
for $i \leftarrow 0$ to $i < LengthOf(data)$ do
    $batch \leftarrow [\,]$
    for $n \leftarrow 0$ to $n < batch\_size$ do
        $image \leftarrow ResizeImage(image)$
        $image \leftarrow NormalizeImage(image)$
        $batch \leftarrow batch + image$
        $n \leftarrow n + 1$
    end for
    $batches \leftarrow batches + batch$
    $i \leftarrow i + n$
end for
$model \leftarrow$ Load pre-trained ResNet-18 CNN model
Remove last layer from $model$
Add layer with randomized weights to $model$
Freeze all layers except last layer of $model$
$lr \leftarrow$ find and store optimal learning rate
for $epoch \leftarrow 0$ to $freeze\_epochs$ do
    for all batch in batches do
        $TrainCNN(model, batch, lr)$
    end for
    $accuracy \leftarrow$ Calculate accuracy of $model$ on validation set
    $loss \leftarrow$ Calculate loss of $model$ on validation set
    $print(accuracy, loss)$
end for
Unfreeze all layers of $model$
$lr \leftarrow$ find and store optimal learning rate
for $epoch \leftarrow 0$ to $unfreeze\_epochs$ do
    for all batch in batches do
        $TrainCNN(model, batch, lr)$
    end for
    $accuracy \leftarrow$ Calculate accuracy of $model$ on validation set
    $loss \leftarrow$ Calculate loss of $model$ on validation set
    $print(accuracy, loss)$
end for
SaveModel($model$)
```

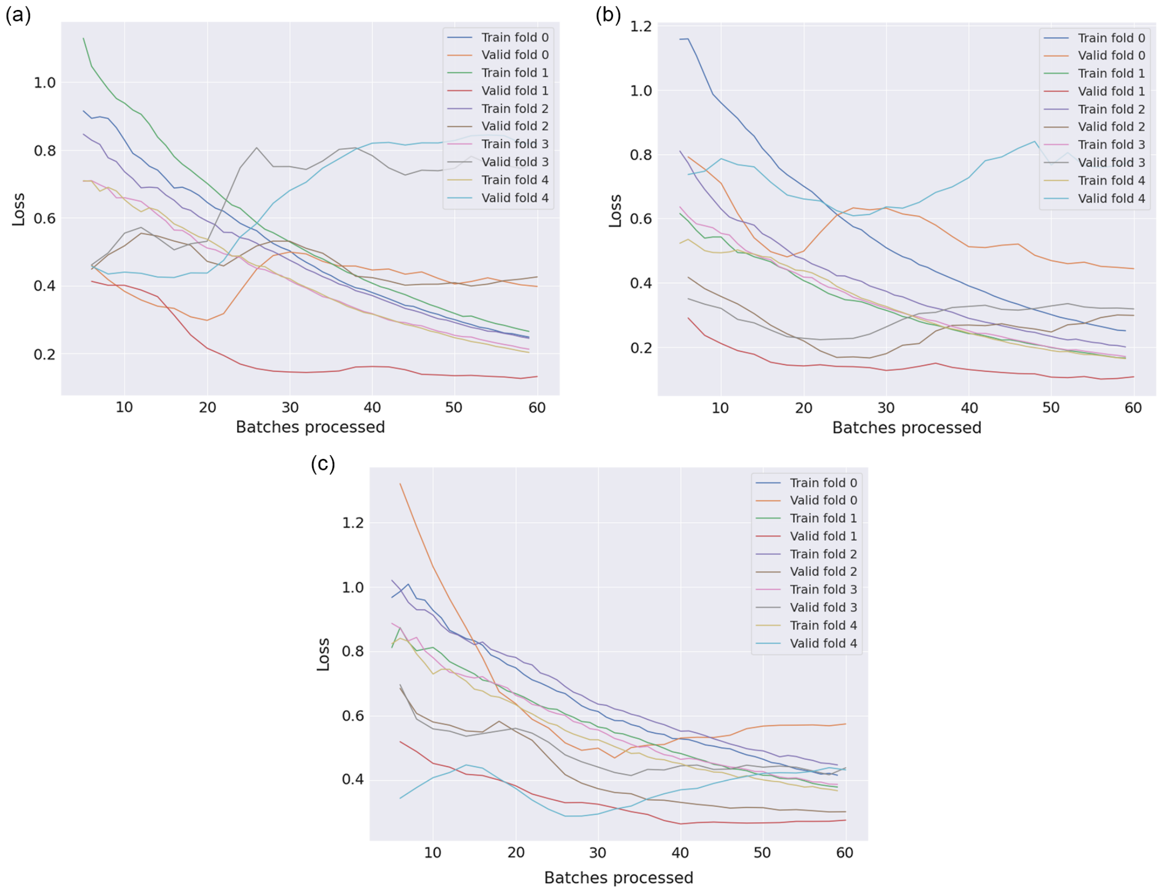
entire dataset and obtaining the average performance metrics allowed for a more accurate depiction of the effectiveness of the suggested techniques.

As for the architecture, an 18-layer pre-trained ResNet-18 model was adopted, which has already been trained to classify, with preassigned weights for each layer. However, the last layer was deleted and replaced with a new layer assigned with random weights because the pre-trained models are specific to the classification task that it was initially created to carry out. This allows the model to be used for the specific tasks relevant to this research. In addition, the Adam optimizer was employed to determine the learning rate, while the cross-entropy loss function was chosen. After creating the model, learning rate finder was utilized to find the optimal learning rate. The model last layer froze at 5 epochs even though with suitable fine-tuning. The model was afterwards trained for an additional 30 epochs after all layers are unfrozen utilizing the learning rate finder as usual to determine optimal learning. The complete algorithm for model generation is shown in Figure 7. The model was run for five times for each of these three models, storing the accuracy and classifications with each run. A cumulative confusion matrix was created after the models had been assessed by averaging the accuracy ratings for all of the partitions to determine the performance metrics, namely true positive rate (TPR), true negative rate (TNR), false positive rate (FPR), and false negative rate (FNR) of each of the models on the overall data. Figure 7 shows model generation algorithm.

4. Evaluation

This section describes the performance metrics and evaluation for the study. The performance metrics are computed at the completion of each new model for the cross-validation models.

Figure 8
Train vs valid loss for original data with (a) no class balancing, (b) resampling, and (c) class weight adjustment



Consequently, there are five series of performance metrics for each of the five cross-validation models, available after model training. To obtain overall performance metrics, averages of the performance scores were computed across the five groups, namely balanced accuracy, precision, recall, F1-score, and AUC-ROC score. In addition, the confusion matrices for all of the models on the validation set were added to create a cumulative confusion matrix. The TPR, TNR, FPR, and FNR for the entire models were equally computed from the confusion matrix.

4.1. Results and discussion

The results consist of six different models trained with various class balancing techniques in an attempt to improve performance. Table 1 summarizes the performance results from all six models. The balanced accuracy, precision, recall, and ROC AUC scores shown are the average of the scores retrieved during the k-fold cross-validation done on each of the models. On the other hand, the TPR, TNR, FPR, and FNR are from the cumulative confusion matrix summed over all of the 5 folds.

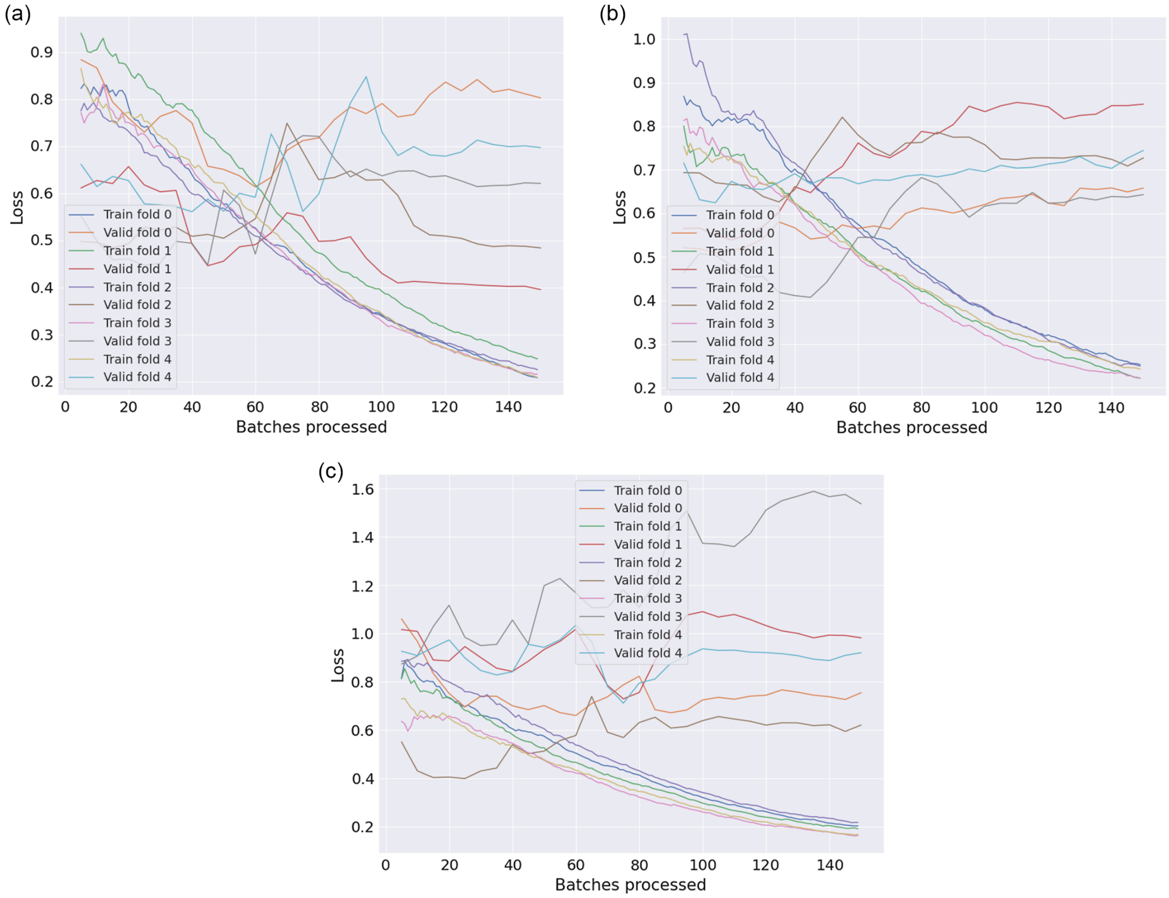
From the results, the baseline model which was trained on the original dataset had the best performance in terms of the balanced accuracy and F1-score, and all of the other models that tried to improve the performance did not yield any better result in terms of these metrics. However, the original model did not have the lowest FNR. The class weight-adjusted model trained on the original data had the lowest FNR, and consequently, the highest recall score among the models tested. In other words, the class weight-adjusted model was least prone to classifying a patient with mortality risk as not having mortality risk. Besides this, the model with random resampling technique applied to the original data had the lowest FPR and highest precision; however, this came at the cost of the lower recall and higher FNR rate. The imputed models, on the other hand, did not perform well in any significant area.

The reason for the models being unable to improve over the baseline may lie in the small size of data that results in the model encountering a lot of unseen data in the validation stage. All these techniques used here for balancing classes do not introduce real variations into the data. Without enough variations in the data, the likelihood of the validation set having unseen data increases greatly. Besides that, the model also fails to generalize real-world data and hence performs poorly when it encounters data that have patterns that it has not encountered before. The imputed dataset despite having more images is not free from this issue. Four hundred seventy-eight images are still small compared to the

Table 1
Train vs validation performance for original data and imputed data

Model		Balanced accuracy	ROC-AUC Score	Precision	Recall	F1 score	True positive rate	True negative rate	False positive rate	False negative rate
Original (169 images)	**Train**	0.92	0.99	0.69	0.99	0.80				
	Test	**0.89**	0.90	0.80	0.84	**0.81**	0.842	0.931	0.069	0.158
Imputed (478 images)	**Train**	0.86	0.93	0.64	0.88	0.73				
	Test	0.74	0.78	0.60	0.60	0.58	0.608	0.864	0.136	0.392
Imputed + class weight adjustment	**Train**	0.89	0.96	0.60	0.96	0.73				
	Test	0.76	0.83	0.54	0.67	0.59	0.67	0.85	0.15	0.33
Imputed + random resampling	**Train**	0.82	0.89	0.82	0.82	0.82				
	Test	0.73	0.80	0.55	0.59	0.56	0.588	0.864	0.052	0.412
Original + random resampling	**Train**	0.91	0.96	0.89	0.94	0.91				
	Test	0.83	0.91	**0.81**	0.71	0.75	0.711	**0.954**	**0.046**	0.289
Original + class weight adjustment	**Train**	0.87	0.97	0.55	0.97	0.70				
	Test	0.88	**0.92**	0.69	**0.90**	0.76	**0.895**	0.863	0.137	**0.105**

Figure 9
Train vs valid loss for imputed data with (a) no class balancing, (b) resampling, and (c) class weight adjustment



number of variations that may be seen in CXRs. On top of that, the imputation could also have been inaccurate, causing the model to get confused as the training sample represents the data incorrectly.

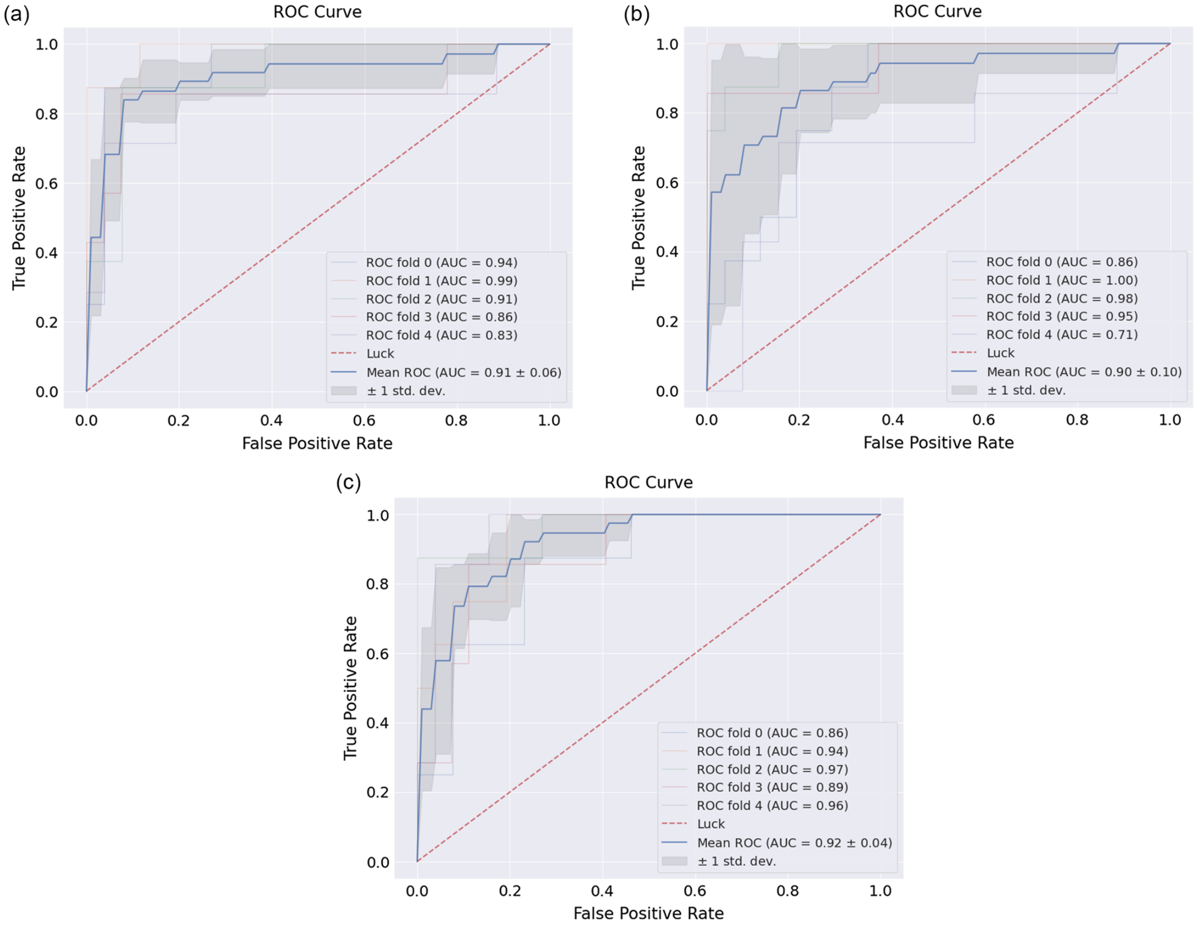
The training vs validation loss plot for each of the models gives a clearer indication of overfitting occurring during training.

Figure 8 and Figure 9 combined the train vs valid loss plot from all the 5 iterations of training during k-fold cross-validation for original data and imputed data respectively. From the figures, it can be noticed that in some cases, the validation loss keeps increasing at every consequent epoch, instead of decreasing. In other words, as the

Table 2
Comparison of ResNet-18 classification performance with existing techniques on the small dataset

Model	Balanced accuracy	ROC-AUC score	Precision	Recall	F1 score	True positive rate	True negative rate	False positive rate	False negative rate
AlexNet [20]	0.76	0.54	0.78	0.96	0.86	0.96	0.12	0.88	0.04
MobileNetV3 [21]	0.79	0.56	0.79	1.00	0.88	1.00	0.12	0.88	0.00
EfficientNetV2 [22]	0.65	0.55	0.79	0.73	0.76	0.73	0.38	0.62	0.27
SwinTransformer [23]	0.74	0.52	0.77	0.92	0.84	0.92	0.12	0.88	0.08
ConvNeXt [24]	0.76	0.50	0.76	1.00	0.87	1.00	0.00	1.00	0.00
ResNet-18	**0.89**	**0.92**	**0.80**	**0.84**	**0.81**	**0.842**	**0.931**	**0.069**	**0.158**

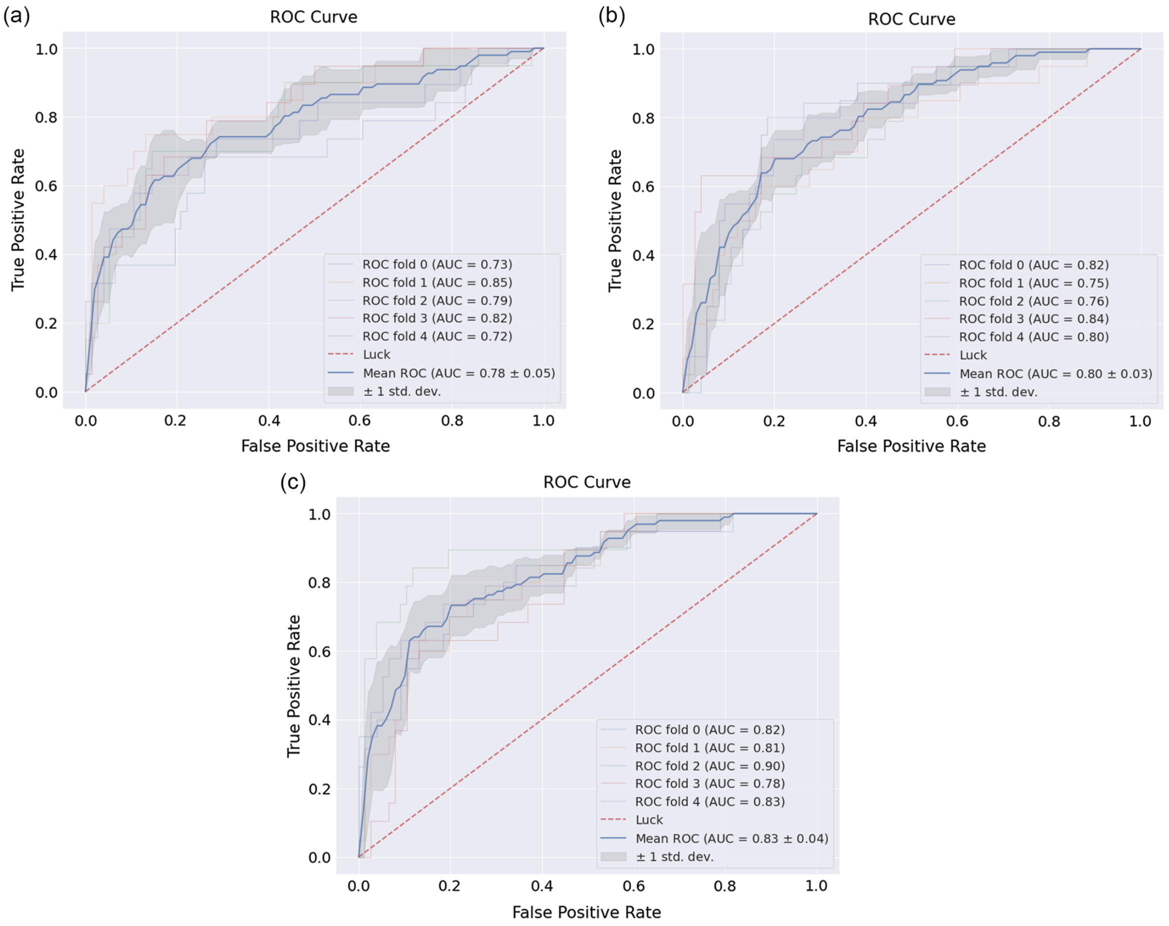
Figure 10
ROC curves for original data with (a) no class balancing, (b) resampling, and (c) class weight adjustment



model learns the data in the training set and adjusts its weights, the model's accuracy and loss in the validation set keep getting worse indicating a lack of correlation between what it is learning during training and what it finds in the validation samples. This happens very frequently in the case of the imputed dataset which is a further indication of having a training set that does not provide an accurate representation of CXR patterns for each of the classes, probably due to mislabeling from inaccurate imputation.

Further analysis of the models was done using the ROC curve and precision–recall (PR) curve. The AUC-ROC scores for each of the

Figure 11
ROC curves for imputed data with (a) no class balancing, (b) resampling, and (c) class weight adjustment



models have been summarized. The ROC and PR curves for the model show the average of the curves generated by the k-fold cross-validation. Figure 10 and Figure 11 show the ROC curves for the original data and imputed data respectively. While Figure 12 and Figure 13 show the PR curve for the original data and imputed data respectively.

The class weight-adjusted model trained on the original data achieves the best average AUC-ROC score of 0.92. The rest of the models do not perform as well, with the model trained on imputed data along with no class balancing performing the worst.

Similarly, the PR curves also show the same trend with the baseline model and class weight-adjusted model on the original data performing the best with an AUC-PR score of 0.79 and the model trained on imputed data along with class weight adjustment performing the worst with an AUC-PR score of just 0.49.

The proposed algorithms based on the ResNet-18 have been compared with the performance of several popular CNN models, including AlexNet [23], MobileNetV3 [24], EfficientNetV2 [25], SwinTransformer [26], and ConvNeXt [27] as shown in Table 2. The evaluation was conducted on the same dataset for image classification task, and the models were compared based on several evaluation metrics including model accuracy, ROC AUC, precision, recall, F1 score, TPR, TNR, FPR, and FNR.

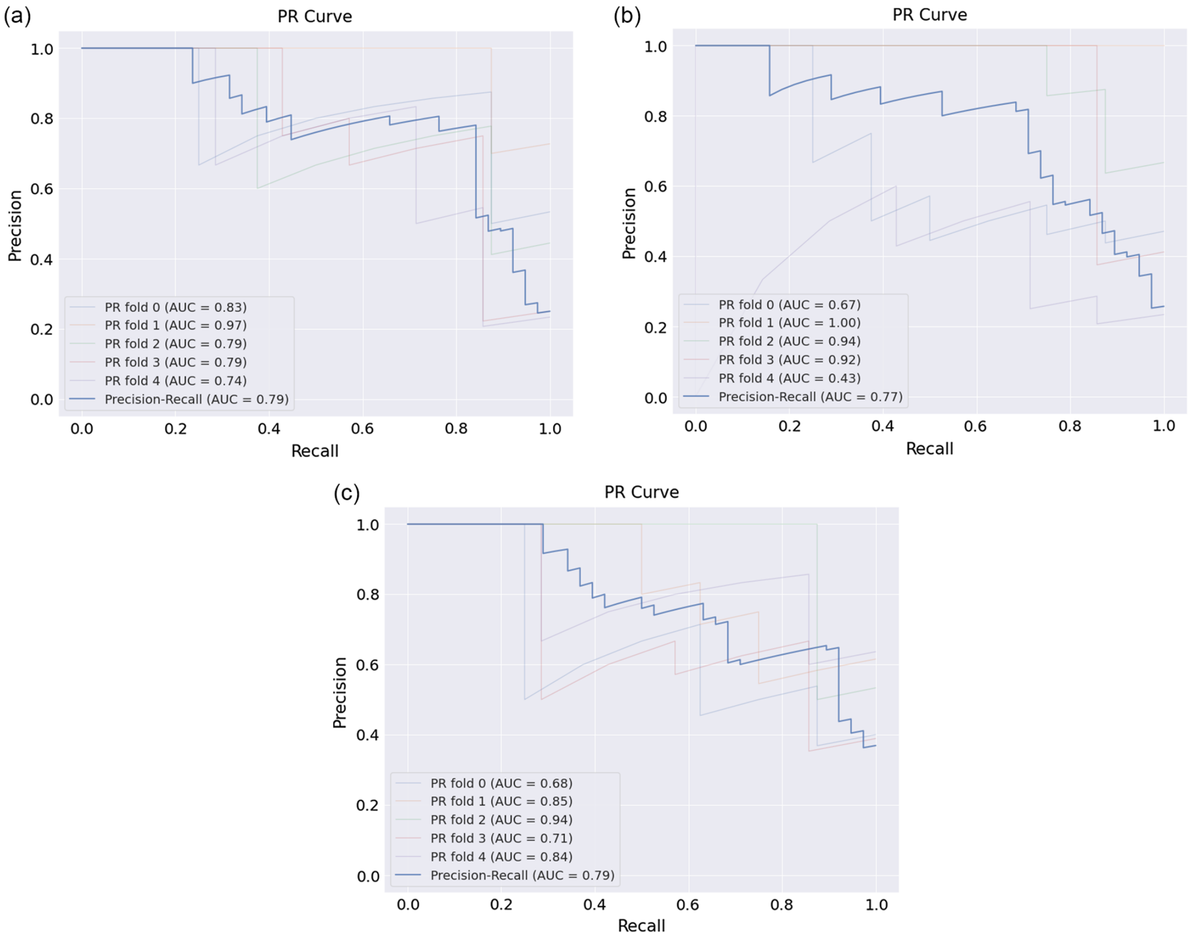
ResNet-18 was used as the baseline model for comparison, and it achieved an accuracy of 0.89, a ROC-AUC of 0.92, a precision of 0.80, a recall of 0.84, and an F1 score of 0.81. The TPR and TNR were 0.842 and 0.931, respectively, while the FPR and FNR were 0.069 and 0.158, respectively.

It is evident that MobileNetV3 has the highest accuracy (0.79) and ROC-AUC score (0.56) among all other trained models, which is still considerably lower in performance compared to the baseline ResNet-18 model. It also has a high precision of 0.79 and a perfect recall of 1.0, indicating that it correctly identified all the positive cases in the dataset. This is an important metric for medical image classification tasks where false negatives can have serious consequences.

AlexNet and ConvNeXt also performed reasonably well with an accuracy of 0.76 and 0.74, respectively. However, their ROC-AUC scores were lower than MobileNetV3, indicating that they may not be as effective in distinguishing between positive and negative cases. EfficientNetV2 had the lowest accuracy of 0.65 and a relatively low ROC-AUC score of 0.55. However, it had a high precision of 0.79, indicating that it correctly identified most of the positive cases in the dataset. SwinTransformer had an accuracy of 0.74 and a relatively low ROC-AUC score of 0.52.

However, it had a high recall of 0.92, indicating that it correctly identified most of the positive cases in the dataset. There are several potential reasons why each model performs the way it does. ResNet-18, as the baseline model, has a relatively simple architecture

Figure 12
PR curves for original data with (a) no class balancing, (b) resampling, and (c) class weight adjustment



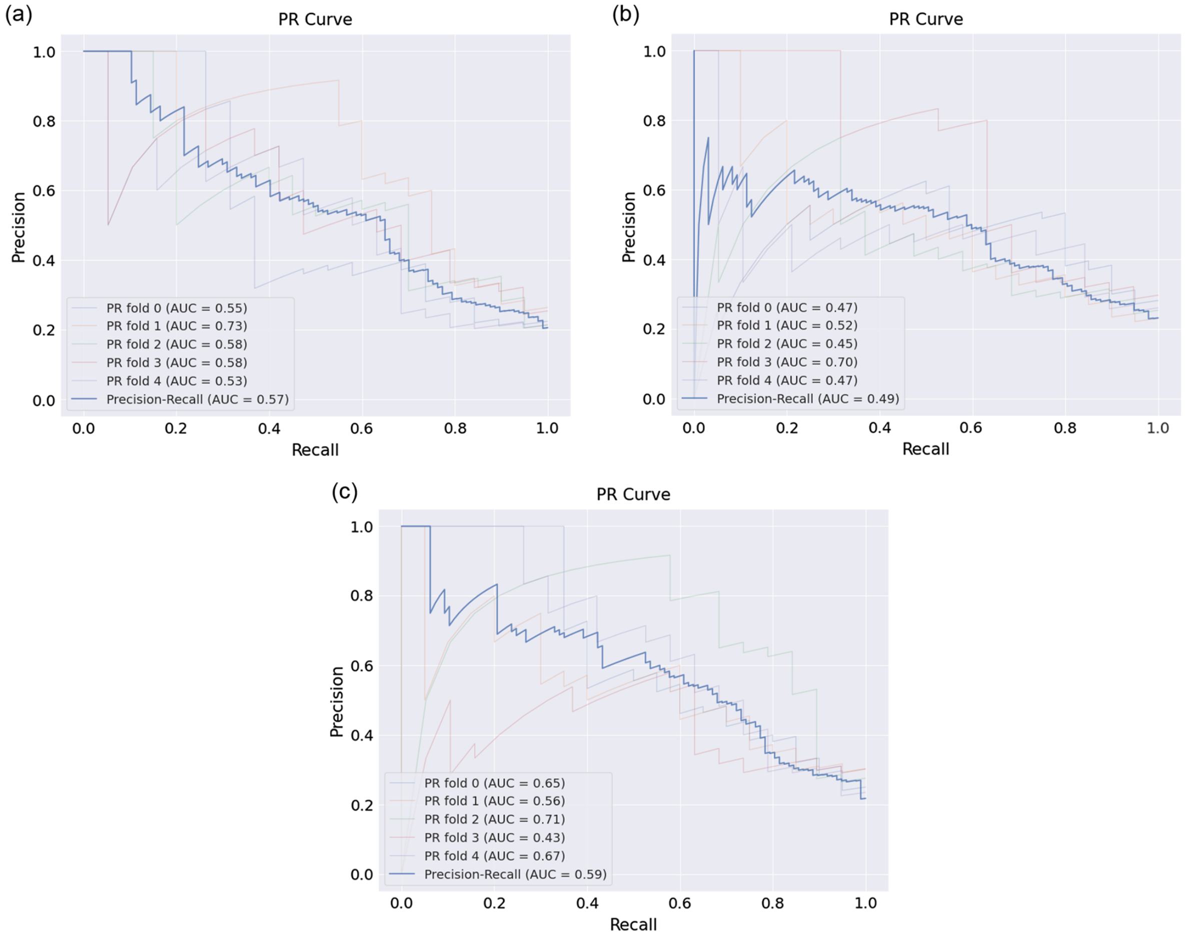
compared to the other models. However, its success in achieving high accuracy and ROC-AUC scores could be attributed to its use of residual connections, which allow for better gradient flow and avoid the vanishing gradient problem.

AlexNet, on the other hand, was one of the earliest CNN models and has a relatively shallow architecture. This may explain its lower performance compared to the other models. Additionally, its lower precision score could be attributed to its tendency to classify some images as positive when they are actually negative. MobileNetV3 is designed to be lightweight and efficient, making it a good option for mobile devices or low-resource environments. Its higher recall score suggests that it is better at identifying true positives than ResNet-18. EfficientNetV2 is designed to be more efficient than its predecessor, EfficientNetV1. However, its lower accuracy and ROC AUC scores suggest that its smaller size comes at the cost of performance. SwinTransformer is a newer model that uses self-attention mechanisms to capture global dependencies in the input image. However, its lower performance compared to ResNet-18 suggests that further optimization may be needed for it to reach its full potential. ConvNeXt uses grouped convolutions to improve efficiency and reduce computational cost. Its high recall score suggests that it is good at identifying true positives, but its lower ROC-AUC score suggests that it may struggle with distinguishing between positive and negative examples.

Overall, each model has its own strengths and weaknesses based on its architecture and design goals. Further optimization or using larger variants of the datasets could potentially improve their performance on the image classification task.

In conclusion, this study showed that ResNet-18 performed well on the image classification task compared to the other CNN models evaluated in this study. However, it is worth noting that the other models could potentially perform better when larger datasets are utilized. The main motivation for using small dataset is due to the onset of COVID-19; data were scarce where only small datasets were available. This study may be a useful contribution in the case of early outbreak of similar epidemic.

Figure 13
PR curves for original data with (a) no class balancing, (b) resampling, and (c) class weight adjustment



5. Conclusion and Future Work

In this paper, we attempted to train deep learning models to predict mortality risk in patients infected with COVID-19 using a small dataset of CXRs images of COVID-19 patients collected from a public GitHub repository. The collected dataset was preprocessed, and two different class balancing techniques were applied on the original and imputed data. The imputed data referred to the use of a RFC model to impute missing labels. Furthermore, a pre-trained ResNet-18 model was fine-tuned on each of the preprocessed data using the FastAI library to produce models with no class balancing, class weight adjustment, and weighted resampling applied for both the original and imputed data. The result showed that the baseline model achieved the best performance in terms of accuracy, but the class weight-adjusted model trained on the original data had the lowest FNR and highest recall and ROC-AUC score. Due to limited data, in our methodology, the CXRs were not discriminated based on the patient's duration of infection at the time of X-ray. Despite that, the result from this research shows the possibility of reliably predicting COVID-19 mortality from CXRs without discriminating based on the patient's duration of infection at the time of X-ray. The performance of the proposed algorithms based on the ResNet-18 compared favorably with those of several popular CNN models, including AlexNet, MobileNetV3, EfficientNetV2, SwinTransformer, and ConvNeXt. This further demonstrates that ResNet-18 is more suited for classification of this nature among the several CNN models considered. This finding is in agreement with previously published studies on low complexity CNNs performance on small datasets such as in Brigato and Iocchi [4].

Of course, when large datasets are available as shown in Olowolayemo et al. [20], experimenting with more pre-trained models of various architectures, utilizing additional X-ray images of COVID-19 cases as well as classifying the cases according to the length of infection, potentially the model may be improved. In modeling X-ray images, larger models frequently produce better accuracy because they should have plenty of variations and patterns to learn. This is not always the case, though. Furthermore, using CXRs of COVID-19 patients taken early after the infection may be more effective for training. It is believed that this should help to determine patient's condition more realistically and indicative of the mortality risk since infection normally worsens with time. Lastly, because of the limited dataset at the onset of COVID-19 pandemic, especially containing a meager number of non-survival or mortality cases, oversampling methods were attempted through a process of image augmentation. This was done to increase size of the datasets, especially for the minority class. While augmenting image is an often utilized technique to increase images' dataset by generating more variations from the images, there is no guarantee that it may adequately account for real-world variations necessary to train an efficient model. It is believed that collection of more X-ray images with considerably increased non-survival cases as well as fine-tuning the models could possibly produce improved performance and efficient results.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data available on request from the corresponding author upon reasonable request.

Author Contribution Statement

Akeem Olowolayemo: Conceptualization, Methodology, Formal analysis, Resources, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration. **Wafaa Khazaal Shams:** Methodology, Formal analysis, Writing – review & editing. **Abubakar Yagoub Ibrahim Omer:** Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Yasin Mohammed:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Visualization. **Raashid Salih Batha:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Visualization.

References

[1] Zhu, W., Wang, Y., Xiao, K., Zhang, H., Tian, Y., Clifford, S. P., . . . , & Huang, J. (2020). Establishing and managing a temporary coronavirus disease 2019 specialty hospital in Wuhan, China. *Anesthesiology*, *132*(6), 1339–1345.

[2] Covid-19: Army opens hospitals to civilians, setting up temporary facilities. (2021). *Times of India*. Retrieved from: https://timesofindia.indiatimes.com/india/covid-19-army-opens-hospitals-to-civilians-setting-up-temporary-facilities/articleshow/82318086.cms

[3] Islam, N., Ebrahimzadeh, S., Salameh, J. P., Kazi, S., Fabiano, N., Treanor, L., . . . , & Cochrane COVID-19 Diagnostic Test Accuracy Group. (2021). Thoracic imaging tests for the diagnosis of COVID-19. *Cochrane Database of Systematic Reviews*. https://doi.org/10.1002/14651858.CD013639.pub4.

[4] Brigato, L., & Iocchi, L. (2021). A close look at deep learning with small data. In *2020 25th International Conference on Pattern Recognition*, 2490–2497. https://doi.org/10.1109/ICPR48806.2021.9412492

[5] Wang, S., Zha, Y., Li, W., Wu, Q., Li, X., Niu, M., . . . , & Tian, J. (2020). A fully automatic deep learning system for COVID-19 diagnostic and prognostic analysis. *European Respiratory Journal*, *56*(2), 2000775. https://doi.org/10.1183/13993003.00775-2020

[6] Kulkarni, A. R., Athavale, A. M., Sahni, A., Sukhal, S., Saini, A., Itteera, M., . . . , & Kulkarni, H. (2021). Deep learning model to predict the need for mechanical ventilation using chest X-ray images in hospitalised patients with COVID-19. *BMJ Innovations*, *7*(2), 261–270. https://doi.org/10.1136/bmjinnov-2020-000593

[7] Luz, E., Silva, P., Silva, R., Silva, L., Guimarães, J., Miozzo, G., . . . , & Menotti, D. (2022). Towards an effective and efficient deep learning model for COVID-19 patterns detection in X-ray images. *Research on Biomedical Engineering*, *38*, 149–162. https://doi.org/10.1007/s42600-021-00151-6

[8] Santa Cruz, B. G., Bossa, M. N., Sölter, J., & Husch, A. D. (2021). Public covid-19 X-ray datasets and their impact on model bias–A systematic review of a significant problem. *Medical Image Analysis*, *74*, 102225. https://doi.org/10.1016/j.media.2021.102225

[9] Cheng, J., Sollee, J., Hsieh, C., Yue, H., Vandal, N., Shanahan, J., . . . , & Wang, J. (2022). COVID-19 mortality prediction in the intensive care unit with deep learning based on longitudinal chest X-rays and clinical data. *European Radiology*, *32*(7), 4446–4456. https://doi.org/10.1007/s00330-022-08588-8

[10] Lee, H. W., Yang, H. J., Kim, H., Kim, U. H., Kim, D. H., Yoon, S. H., . . . , & Goo, J. M. (2023). Deep learning with chest radiographs for making prognoses in patients with COVID-19: Retrospective cohort study. *Journal of Medical Internet Research*, *25*, e42717. https://doi.org/10.2196/42717

[11] Munera, N., Garcia-Gallo, E., Gonzalez, Á., Zea, J., Fuentes, Y. V., Serrano, C., . . . , & Reyes, L. F. (2022). A novel model to predict severe COVID-19 and mortality using an artificial intelligence algorithm to interpret chest radiographs and clinical variables. *ERJ Open Research*, *8*(2), 00010-2022. https://doi.org/10.1183/23120541.00010-2022

[12] Islam, M., Hannan, T., Sarker, L., & Ahmed, Z. (2022). COVID-DenseNet: A deep learning architecture to detect COVID-19 from chest radiology images. *Preprints*. https://doi.org/10.20944/preprints202005.0151.v3

[13] Wang, T., Nie, Z., Wang, R., Xu, Q., Huang, H., Xu, H., . . . , & Liu, X. J. (2023). PneuNet: Deep learning for COVID-19 pneumonia diagnosis on chest X-ray image analysis using vision transformer. *Medical & Biological Engineering & Computing*, *61*, 1395–1408. https://doi.org/10.1007/s11517-022-02746-2

[14] Aslani, S., & Jacob, J. (2023). Utilisation of deep learning for COVID-19 diagnosis. *Clinical Radiology*, *78*(2), 150–157. https://doi.org/10.1016/j.crad.2022.11.006

[15] Azad, A. K., Mahabub-A-Alahi, Ahmed, I., & Ahmed, M. U. (2023). In search of an efficient and reliable deep learning model for identification of COVID-19 infection from chest X-ray images. *Diagnostics, 13*(3), 574. https://www.mdpi.com/2075-4418/13/3/574#

[16] Constantinou, M., Exarchos, T., Vrahatis, A. G., & Vlamos, P. (2023). COVID-19 classification on chest X-ray images using deep learning methods. *International Journal of Environmental Research and Public Health*, *20*(3), 2035. https://doi.org/10.3390/ijerph20032035

[17] Nayak, S. R., Nayak, D. R., Sinha, U., Arora, V., & Pachori, R. B. (2023). An efficient deep learning method for detection of COVID-19 infection using chest X-ray images. *Diagnostics*, *13*(1), 131. https://doi.org/10.3390/diagnostics13010131

[18] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

[19] Saltz, J., Saltz, M., Prasanna, P., Moffitt, R., Hajagos, J., Bremer, E., . . . , & Kurc, T. (2021). *Stony Brook University COVID-19 positive cases* [Data set]. The Cancer Imaging Archive. https://doi.org/10.7937/TCIA.BBAG-2923.

[20] Olowolayemo, A., Yasin, M., & Salih, M. R. (2023). Predicting mortality risk of COVID-19 patients using chest X-rays. *International Journal on Perceptive and Cognitive Computing*, *9*(1), 33–43.

[21] Cohen, J. P., Morrison, P., Dao, L., Roth, K., Duong, T. Q., & Ghassemi, M. (2020). Covid-19 image data collection:

Prospective predictions are the future. *arXiv Preprint: 2006.11988*.

[22] Kokla, M., Virtanen, J., Kolehmainen, M., Paananen, J., & Hanhineva, K. (2019). Random forest-based imputation outperforms other methods for imputing LC-MS metabolomics data: A comparative study. *BMC Bioinformatics*, *20*(1), 492. https://doi.org/10.1186/s12859-019-3110-0

[23] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

[24] Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., . . . , & Adam, H. (2019). Searching for MobileNetV3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1314–1324.

[25] Tan, M., & Le, Q. (2021). EfficientNetV2: Smaller models and faster training. In *Proceedings of the 38th International Conference on Machine Learning*, *139*, 10096–10106.

[26] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., . . . , & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.

[27] Liu, Z., Mao, H., Wu, C. Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11976–11986.

How to Cite: Olowolayemo, A., Shams, W. K., Omer, A. Y. I., Mohammed, Y., & Batha, R. S. (2025). COVID-19 Mortality Risk Prediction Using Small Dataset of Chest X-ray Images. *Artificial Intelligence and Applications*, *3*(1), 94–109. https://doi.org/10.47852/bonviewAIA3202819