MethodsX 11 (2023) 102295

Contents lists available at ScienceDirect

MethodsX

journal homepage: www.elsevier.com/locate/methodsx

Classification of COVID-19 X-ray images using transfer learning with visual geometrical groups and novel sequential convolutional neural networks

Sunil Kumar[a,b,*], Harish Kumar[b]

[a] *Department of Information Technology, School of Engineering and Technology (UIET), CSJM University, Kanpur, UP, India*
[b] *Department of Computer Engineering, J.C. Bose University of Science and Technology, YMCA, Faridabad, India*

ARTICLE INFO

Method name:
Novel sequential CNN in contrast to transfer learning using VGG16 and VGG19

Keywords:
Deep learning
LBP
Machine learning
SVM
VGG16
VGG19

ABSTRACT

COVID-19 is a highly transmissible infectious disease that remains a substantial challenge. The utilization of chest radiology, particularly X-ray imaging, has proven to be highly effective, easily accessible, and cost-efficient in detecting COVID-19. A dataset named COVID-Xray-5k, consisting of imbalanced X-ray images of COVID-19-positive and normal subjects, is employed for investigation. The research introduces a novel methodology that utilizes conventional machine learning (ML), such as local binary patterns (LBP) for feature extraction and support vector machines (SVM) for classification. In addition, transfer learning is employed with the Visual Geometry Group 16-layer (VGG16) and 19-layer (VGG19) models. Besides, novel sequential convolutional neural network (CNN) architectures are presented to develop an autonomous system for classifying COVID-19. One of the proposed CNN architectures classifies the test dataset with an F1 score of 91.00% and an accuracy of 99.45% based on an empirical investigation to determine optimal hyper-parameters. The methods presented in the research show promising potential for COVID-19 classification, irrespective of class imbalance.

- Employment of ML models to investigate subjective feature engineering and classification.
- Transfer learning was employed for VGG16 and VGG19 with eight distinct models.
- Illustration of two novel CNN sequential architectures; all the investigation is performed with and without weighted sampling.

Specifications table

Subject area	Computer Science
More specific subject area	Machine Learning and CNN
Name of your method	Novel sequential CNN in contrast to transfer learning using VGG16 and VGG19
Name and reference of original method	Not applicable – The methods that are being presented make use of several different methods, each of which is addressed and mentioned in the article that is being used as a reference.
Resource availability	Dataset Link - https://github.com/shervinmin/DeepCovid/tree/master/data

[*] Corresponding author.
E-mail address: sunilymca1@jcboseust.ac.in (S. Kumar).

https://doi.org/10.1016/j.mex.2023.102295
Received 14 December 2022; Accepted 20 July 2023
Available online 22 July 2023

Introduction

The World Health Organization (WHO) issued an official statement in March 2020 stating that the coronavirus disease 2019 (COVID-19) epidemic had turned into a global pandemic [1]. To effectively combat COVID-19, it is necessary to identify the presence of severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) in the lungs, as this virus is the root cause of the disease. The reverse transcription-polymerase chain reaction, often known as RT-PCR, is the nucleic acid amplification test (NAAT) employed mainly in detecting SARS-CoV-2. Despite its specificity, the process is intricate and requires significant time, with a turnaround time of up to 48 h; the process in issue is often seen negatively [2]. The RT-PCR testing methodology exhibits several limitations, including its intricate and prolonged nature.

In contrast to the relatively high cost and limited accessibility of RT-PCR, X-ray imaging is notable for its widespread availability and cost-efficient in nearly all healthcare facilities. Numerous studies have demonstrated the effectiveness of X-ray scans in detecting COVID-19, illustrating its potential [3]. Furthermore, the results of X-ray scans can be obtained expeditiously, rendering them a highly accessible diagnostic tool [4]. Computed tomography (CT) scans are relatively more expensive than X-ray scans and potentially unaffordable for several people. When juxtaposed with X-rays, picking up CT scans is considerably more challenging [5].

To attain broad generalizability, ML models must account for the heterogeneity of COVID-19 forms and X-ray characteristics across diverse populations and geographic regions. ML is a resilient area of research that empowers machines (computers) to acquire knowledge and generate forecasts without the need for explicit programming. One fundamental aspect of ML is subjective feature engineering, which involves creating, transforming, and selecting significant features from raw data to improve the performance of ML classifiers. With the aid of ML in healthcare, Specifically, ML algorithms have been used to design a pipeline for the task of classification [6]. In classification, an ML classifier is trained using labeled data, and then this trained classifier is used to predict class labels for new samples. LBP, a feature descriptor and a widely employed technique in feature extraction, is particularly effective in discriminating various textures in the image [7]. It extracts texture information by comparing the intensity of each pixel with its surrounding neighbors and encoding these comparisons into binary patterns. The method generates a histogram of local patterns, resulting in a concise feature representation [8]. The histogram's dimensionality may be further decreased using methods such as Principal Component Analysis (PCA), which aids in regulating the curse of dimensionality and computing complexity. It is a widely used technique for feature transformation that involves projecting high-dimensional data onto a lower-dimensional space while retaining as much of the original data variation as possible [9]. PCA achieves this by identifying the principal components that capture the most significant patterns in the data [10]. PCA executes a linear transformation to achieve a correlation between the LBP features. The process generates a novel compilation of principal components that exhibit no correlation and are arranged according to their significance. The method of correlation aids in the elimination of insignificant data and facilitates the identification of the most salient features for classification [11]. SVM is a popular ML classifier for binary classification, focusing on identifying optimal hyperplanes. It maximizes margins and handles high-dimensional data and decision boundaries using kernel functions [12].

Deep learning (DL) models, renowned for their capacity to acquire intricate patterns from extensive datasets, have emerged as among the most promising approaches for detecting COVID-19. CNNs are a specialized form of DL that focuses on image classification [13]. A wide range of pre-trained models available in the field of CNN can be effectively utilized for this purpose by making straightforward modifications. MobileNet, Residual Neural Network (ResNet), DenseNet, VGG16, and VGG19 are prominent pre-trained models commonly employed for the specified task. The MobileNet architecture is a CNN characterized by its lightweight shape, enabling it to achieve high accuracy while utilizing fewer computational resources. By employing depth-wise separable convolutions, MobileNet achieved impressive reductions in model size and inference time [14]. The ResNet gained considerable attention because of its ability to effectively address the problem of vanishing gradients that often arise in deep neural networks. Utilizing a hierarchical structure in ResNet facilitated the identification and analysis of complex patterns, rendering it well-suited for the precise classification of X-ray images [15]. The Densenet architecture is recognized for its dense connectivity patterns, which facilitate the effective propagation of features across different layers. Including densely connected layers in the network enables the acquisition of local and global features, thereby augmenting the model's ability to discern and differentiate between various patterns and characteristics [16]. The CNN-IFO method was employed, which combines the weighted mean of vector optimizer (INFO) with CNN models [17]. It was done to improve the accuracy of predictions.

Time series modeling is a statistical methodology employed for analyzing and predicting data that is systematically gathered at periodic intervals, and It can offer valuable insights and enhance forecast accuracy by considering the data's temporal dependence and other characteristics. A recurrent neural network (RNN) represents one of the available models that can be employed to address the complexities associated with time series analysis. Long-short-term memory (LSTM) networks, a type of RNN, are good at modeling and analyzing sequences for the task [18]. The LSTM and Ant-Lion Optimizer (LSTM-ALO) algorithm was employed to optimize the number of hidden layer neurons and the learning rate [19]. Utilizing the benefits of ALO, researchers used it to improve the initial weights and thresholds of LSTM networks. It made LSTM networks more accurate and better [20].

Transfer learning is a technique that leverages pre-trained models on extensive datasets to expedite the training process and enhance the ability of models to perform well on new, unseen data. This process involves identifying and extracting pertinent features and patterns from closely related tasks, thereby enhancing the effectiveness and efficiency of transfer learning [21]. Transfer learning has been used on the ImageNet dataset using the VGG16 and VGG19 models. Both visual geometrical groups were pre-trained, and models were tried out to save time and computation [22,23]. Transfer learning, which took the form of an extractor-classifier pair, resulted in effective detection. Both the MobileNet architecture with the SVM and the DenseNet201 architecture with the Multi-Layer Perceptron (MLP) classifier were able to perform better [14].

Our research aims to create an ML model for classifying X-ray scans of patients using an imbalanced dataset such as COVID-Xray-5k. The framework integrates LBP for texture description, PCA for dimensionality reduction, and an SVM classifier for COVID-19 image classification. Once ML methods are employed to examine the process of subjective feature engineering and classification, the next step in the investigation is to move on to DL methods, as the research revealed that the performance of the ML models is subpar. Employing the transfer learning techniques VGG16 and VGG19 with eight different models each, two other pipelines are built to diagnose based on the given data, with and without weighted sampling. The objective was to decrease the number of trainable parameters while maintaining performance. A notable enhancement in the performance of VGG16 was observed in the investigation. The VGG16 model is noted for its high computational cost during training, which can be attributed to the considerable number of features it employs.

Additionally, the model's interpretability is relatively low, and using a dataset with a distinctive image distribution for prediction may impact how well it performs. The investigational findings presented by the VGG16 and VGG19 CNN architectures necessitate devising novel sequential CNN architectures for performance improvement. The research proposes two novel sequential CNN models with and without dropout layers and weighted sampling, then tunes Hyperparameters for the task. Our methodology uses ML and its subfields to develop a diagnostic method that helps automate the jobs in the classification of COVID-19. The proposed methodologies can also facilitate the detection of crucial COVID-19 variants when a sufficiently large dataset is supplied. The research assesses the performance of the ML model, utilizes transfer learning with VGG16 and VGG19, and then develops sequential CNN models. It also discusses the challenges experienced when working with ML, its subfields, and how they were addressed.

Methods

Machine learning

Image preprocessing

Variations in X-ray images can be attributed to several factors, such as dissimilarities in imaging equipment, imaging procedures, patient orientation, and the extent and advancement of the ailment. The presence of discrepancies in image quality and visual attributes may pose challenges for ML models to acquire uniform patterns to detect COVID-19. The image preprocessing procedure encompassed essential steps such as converting to grayscale and resizing [24].

Feature engineering

Feature extraction aims to identify the most informative and pertinent features for the given problem while simultaneously reducing data dimensionality and eliminating noise. The features that have been extracted possess the potential to serve as input for an ML classifier [25]. The LBP technique is a widely recognized approach for feature extraction from images.

LBP

LBP transforms the raw input data into a new set of features that are more informative and representative of the underlying patterns in the data [7]. When determining the features of an image, the weighted average value of nearby pixels—which depends on how intense those pixels are concerning the center pixel—replaces each pixel in the image [26]. The basic idea behind LBP is to compare the intensity of a pixel with its surrounding pixels and encode the result as a binary number. This results in a texture descriptor invariant to monotonic gray-scale changes and sensitive to the local structure of the image. This process is repeated for each pixel in the image, resulting in a feature vector for the entire image. LBP can be expressed mathematically:

$$LBP(p) = \sum_{i=0}^{7} T(I(p) - I(p+r))\, 2^{i} \tag{1}$$

Where *p* is the central pixel, *i* is the index of the surrounding pixel, *T(x)* is a function that returns 1 if $x > 0$ and returns 0 if $x <= 0$, *I(p)* is the intensity of the central pixel, and *I(p + r)* is the intensity of the surrounding pixel, located at a distance of *r* from the central pixel [27].

Standard scaling

Following the completion of the feature extraction process and the acquisition of a set of transformed features, it is common to implement standard scaling as a preliminary step. Data preprocessing facilitates efficient and robust model training by enhancing speed and stability. Standard scaling, also known as Z-score normalization, involves subtracting the mean and dividing by the standard deviation of each feature presented through Eq. (2). This process ensured that each feature had a mean of 0 and a standard deviation of 1, making them comparable and bringing them to a similar scale [28]. It ensures that all the features are treated equally in scale and distribution [29].

$$z = \frac{(x - \mu)}{s} \tag{2}$$

z = transformed data, x = data sample, μ = mean, s = standard deviation.

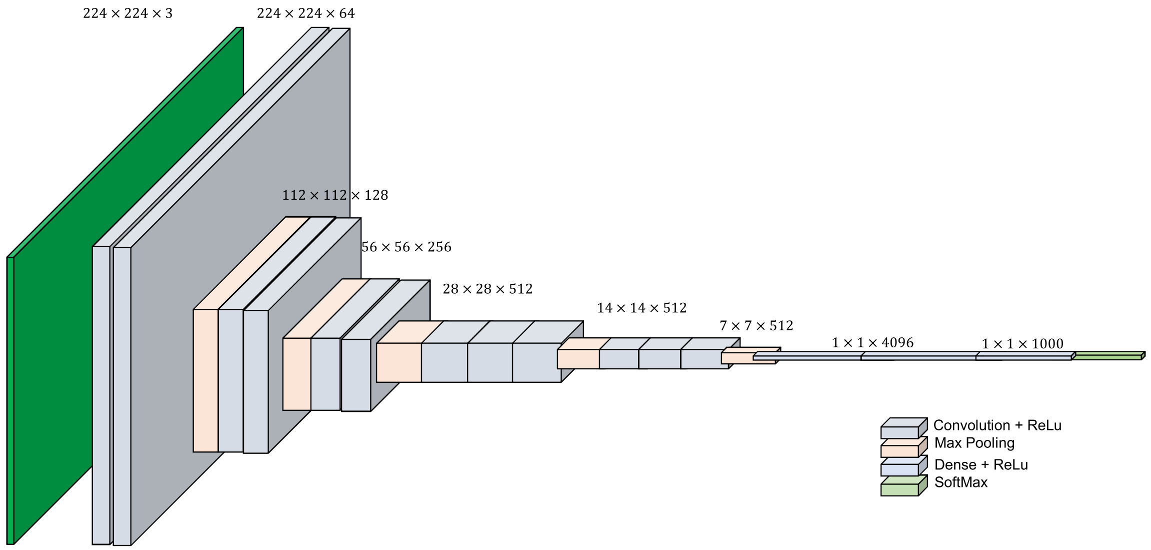

Fig. 1. VGG16 Architecture.

PCA

PCA is a dimensionality reduction approach that transforms scaled data by projecting it onto a new set of orthogonal axes known as principal components. The purpose is to maximize the variance of the data across the first few main components while limiting data loss. It can be performed by constructing a scatter matrix and determining the eigenvectors. These vectors may aid in the extraction of reduced data. It can identify the most basic patterns in the data while accounting for the relative variances of the features [10,9].

Weighing

Class weight is a technique used to assign different weights to classes in a classification task to address the class imbalance and improve the ML model's performance on minority classes. The loss function is altered to compensate for the imbalanced number of samples among classes, and a higher weight is assigned to the loss encountered by samples associated with the minority class [30]. Eq. (3) provides the class weights for the individual classes [28].

$$\mathrm{class}_{\mathrm{weight}} = \frac{\mathrm{n}_{\mathrm{samples}}}{\left(\mathrm{n}_{\mathrm{classes}} * \mathrm{f}_{\mathrm{c}}(\mathrm{y})\right)} \tag{3}$$

Where, $\mathrm{class}_{\mathrm{weight}}$ = weight assigned to each class, $\mathrm{n}_{\mathrm{samples}}$ = total number of samples in the dataset, $\mathrm{n}_{\mathrm{classes}}$ = number of classes, f_{c} = frequency count of each class.

SVM classifier

SVM employs support vectors to identify a separating hyperplane in a multi-dimensional space. The hyperplane is optimized for maximum margin and serves as a decision boundary for classifying data samples. The support vectors, which are data samples close to the hyperplane, have a significant impact on the placement and alignment of said hyperplane. The SVM algorithm enables automated data mapping to a more intricate feature space by utilizing kernel functions such as the radial basis function (RBF) [31].

CNN

The field of DL employs a methodology known as convolution to derive prominent features from input images. The process entails the utilization of convolution kernels or filters that possess shared weights to generate feature maps that exhibit shift invariance [32]. A CNN is a specific type of neural network that uses convolutional operations in at least one of its hidden layers to generate a feature map from the input matrix for that layer, which acts as the input for the next layer [33]. Other layers, such as pooling layers, are generally added after convolution layers, which help reduce data dimensions, and fully connected layers, similar to traditional MLP [34].

VGG16 and VGG19

Based on their unique architectures, the VGG16 and VGG19 models have 16 and 19 layers of convolutional and pooling layers, respectively, followed by three fully connected layers. VGG-19′s layout is similar to VGG-16′s, except for further convolution layers. Because of their consistent performance and well-established structural design, both models have gained widespread recognition as conventional standards for assessing the utility of novel CNN models. The proliferation of research endeavors in computer vision can be attributed to the widespread adoption of CNNs, with VGG16 and VGG19 being recognized as fundamental models that reinforce the field [23]. The VGG16 model's comprehensive block diagram is illustrated in Fig. 1.

Transfer learning

In the field of DL, it is frequently suggested to utilize a previously trained model in instances where there is inadequate training data to train a model with random initial weights properly. Utilizing pre-trained models to solve new tasks is known as "transfer learning." The pre-trained model underwent adaptation to suit a distinct yet interconnected task. The pre-trained convolution layers are employed as a feature extractor, and the top dense layers are replaced with varying output layers for fine-tuning the task [21,33].

Transfer learning with VGG16 and VGG19

VGG16 and VGG19 are trained on the ImageNet dataset, which contains millions of labeled images from various categories. These models have learned to extract significant features, making them outstanding choices for transfer learning. In the context of transfer learning, two prevalent scenarios are highlighted: one that involves class weight and another that does not. Transfer learning is used when already trained models, like VGG16 or VGG19, are used as feature extractors without class weight. The pre-trained model's weights are restrained, and only the final layers of the network are adjusted and trained on the intended dataset. As mentioned earlier, the methodology enables the model to exploit the acquired representations from ImageNet while refining the ultimate layers to conform to the particular task in the issue [21–23].

In contrast, using transfer learning in conjunction with class weight entails not only the adaptation and refinement of the ultimate layers but also considering class imbalances in the intended dataset. Class weight pertains to allotting greater weights to underemphasized classes during training. As mentioned earlier, the modification facilitates the model by mitigating the issue of prejudiced training that arises due to uneven class distributions, thereby enhancing overall efficiency and minimizing the influence of class imbalances [35]. The utilization of transfer learning through VGG16 or VGG19 can establish a robust basis for extracting intricate image features, thereby expediting the creation of models that exhibit superior performance.

Sequential cnn architecture

Sequential CNN architecture encompasses various vital characteristics, such as hierarchical feature learning, feature extraction, invariance, and transfer learning. The factors mentioned earlier facilitate the ability of CNNs to acquire knowledge and recognize characteristics at various scales and levels of abstraction. Sequential CNNs use convolutional layers to represent input data hierarchically, capturing local patterns and traits. These layers abstract features into deeper representations, helping the model understand complex visual structures. These CNNs can identify features at multiple sizes and abstraction levels, enabling fast feature extraction and representation learning. Sequential CNNs are translational invariant, allowing them to recognize patterns in the input image regardless of spatial location. Shared weights and pooling processes help CNNs catch and retain significant features. It will enable CNNs to manage translations and input data variances. Researchers may use the stored information in these pre-trained CNNs by fine-tuning them for individual datasets. Even with insufficient training data, this strategy helps them perform well. This capability diminishes the likelihood of overfitting and enhances the effectiveness of processing extensive datasets [36]. The inherent characteristics of sequential CNN models motivate us to explore novel architectural designs. In contemplating this, two distinct sequential CNN architectures have been developed and put forth for application in the classification of COVID-19.

Proposed methods

First proposed method

Within the scope of the proposed model 1 provided (Fig. 2), a 12-layer CNN model was utilized as the initial approach.

The 12-layer network comprises convolutional layers, max-pooling layers, flattening layers, dense layers, and activation functions. The convolutional layer is the area of learning in the model and the site of the convolution process. The system consists of filters and kernels. Convolutional layers employ filters to extract features and generate a feature map. The method utilized 64, 128, and 256 filters for the initial, second, and third convolutional layers, with 11×11, 5×5, and 3×3 kernel sizes. The initial approach involved utilizing three conv2D (2-D convolution) layers. In cases where an image is rendered to an extent that exceeds the optimal dimensions, a pooling layer is used to reduce its volume. The Max Pooling Layer is widely recognized as a prominent pooling layer. The maximum pooling operation selects the highest value within the feature map for further processing. A Maxpool2D layer was utilized for every Conv2D layer in our study.

The pooled feature map matrix is condensed into a single column using a flattened layer. The dense layer is responsible for identifying the features that exhibit the highest similarity with a particular class. The first and second dense layers comprised a densely connected layer consisting of 1024 neuron units, whereas the third dense layer utilized 512 neurons. The Rectified Linear Unit (ReLu) activation function mitigates the computational burden across various layers. The Softmax activation function is used to classify the outputs.

Second proposed method

The second proposed method entailed employing a 13-layer CNN model, as illustrated in the provided model 2 (Fig. 3). The architecture of this network is similar to Model 1, except for including a dropout layer. The dropout technique is employed during training to exclude or disregard neuron units. This measure serves to alleviate the problem of overfitting in the model. It is possible to achieve the desired outcome by probabilistically preserving each node in a layer's output with an estimated probability of 0.5.

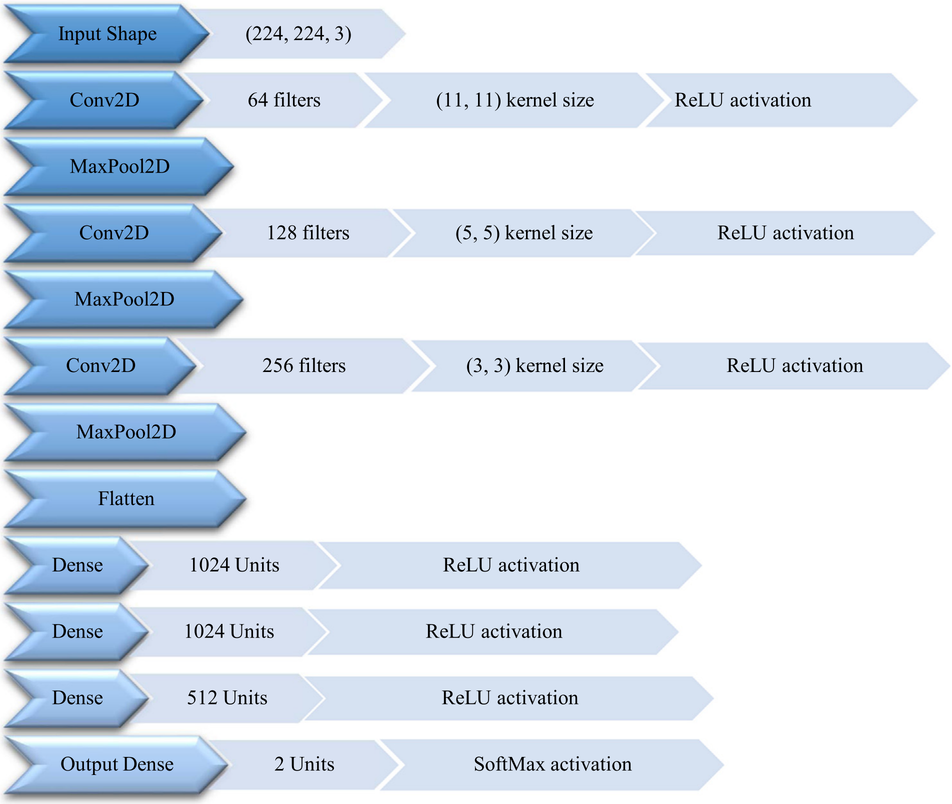

Fig. 2. Proposed Model-1.

Performance metrics

Assessing performance is a fundamental aspect of determining whether improvements are being achieved. The utilization of imbalanced data across classes may lead to accuracy scores that are satisfactory but ultimately misleading, as noted in reference [37]. The following metrics are utilized for the assessment of pipelines:

Accuracy

It is the proportion of accurately labeled predictions to overall predictions. It can be from 0 percent to 100 percent.

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \tag{4}$$

Specificity

It is the proportion of accurately negative-labeled samples among all those that are negative.

$$Specificity = \frac{TN}{(TN + FP)} \tag{5}$$

Sensitivity

It is the proportion of appropriately positive-labeled samples to all positive samples.

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

Precision

It is the proportion of positively labeled samples that are accurately labeled among all positively labeled samples.

$$Precision = \frac{TP}{(TP + FP)} \tag{7}$$

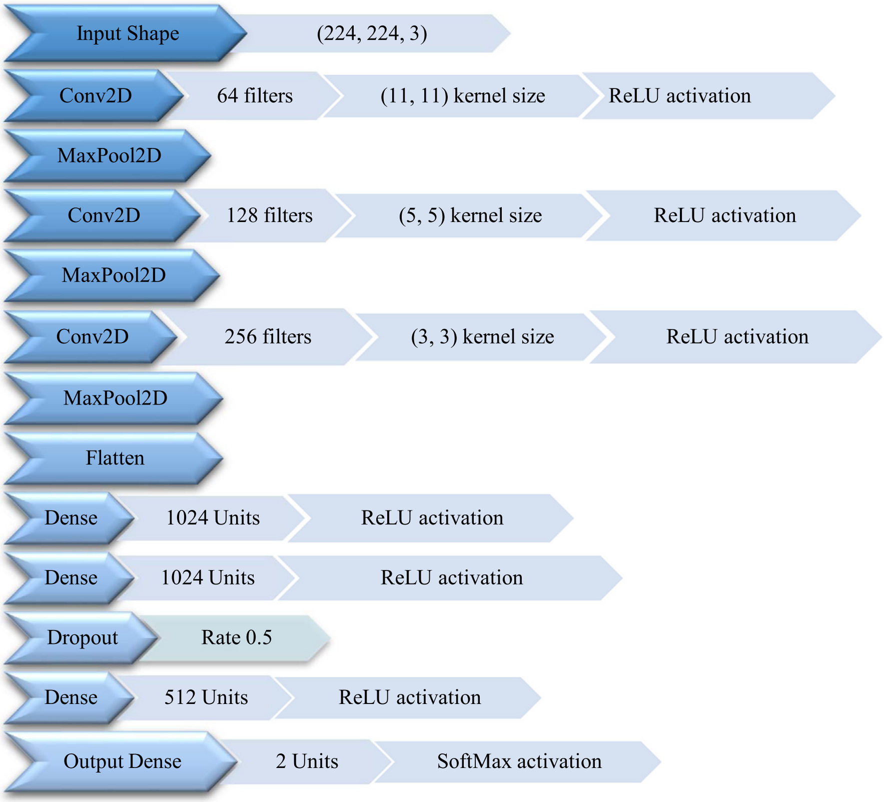

Fig. 3. Proposed Model-2.

Table 1
Confusion Matrix.

		Predicted Class	
		Negative	Positive
Actual Class	Negative	TN	FP
	Positive	FN	TP

F1-Score

The F1-score is a form of the Pythagorean mean, which is the harmonic mean of the precision and recall values [39].

$$\text{F1 Score} = 2* \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} \ + \text{Recall})} \tag{8}$$

Confusion matrix

It visualizes the performance of a classification algorithm and forms the basis for other metrics. For a binary classification problem, the confusion matrix [33] is shown in Table 1.

In Table 1, The number of cases in which both the real labels and the estimated labels are positive is referred to as the true positive (TP). The number of cases in which both the real labels and the estimated labels are negative is represented by the true negative, abbreviated as TN. The number of cases in which the real label was negative but the estimated label was positive is referred to as the number of false positives (FP). Finally, the number of cases in which the real label was positive even though the estimated label was negative is referred to as the false negative (FN) [38].

Materials and method validation

The principal objective of the research is to develop a methodical framework for classifying imbalanced COVID-19 data. ML techniques, transfer learning with the VGG16 and VGG19 models, and new sequential CNN architectures that used both weighted and unweighted sampling techniques were all used in the research.

Table 2
Dataset Distribution.

Split	Positive	Negative	Total
Train	84	2000	2084
Test	100	3000	3100
Total	184	5000	5184

Table 3
ML Pipeline.

Pipeline	Feature Extraction	Feature Transform	Classifier	Class Weights
First	LBP (default)	–	SVM	{0: 1, 1: 1}
Second	LBP (default)	PCA	SVM(RBF)	{0: 1, 1: 1}
Third	LBP (uniform)	–	SVM	{0: 1, 1: 1}
Fourth	LBP (uniform)	PCA	SVM(RBF)	{0: 1, 1: 1}
Fifth	LBP (default)	–	SVM	{0: 0.521, 1: 12.40}
Sixth	LBP (default)	PCA	SVM(RBF)	{0: 0.521, 1: 12.40}
Seventh	LBP (uniform)	–	SVM	{0: 0.521, 1: 12.40}
Eighth	LBP (uniform)	PCA	SVM(RBF)	{0: 0.521, 1: 12.40}

COVID-Xray-5k dataset

The COVID-Xray-5k dataset is a publicly available dataset used in our method. The COVID-Xray-5k dataset was generated from chest X-ray images from two different datasets: The Covid-Chestxray-Dataset, which included COVID-19 X-ray samples, and the ChexPert dataset, which had non-COVID samples. The database comprises chest radiology scans for both subjects with and without COVID-19. Shervin Minaee et al. (https://github.com/shervinmin/DeepCovid/tree/master/data) planned and assembled the dataset [4]. This dataset helps in the prognosis and diagnosis of COVID-19 in humans and includes 3100 test images in addition to the 2084 training images. In our investigation, the input image data was split to form a new training and validation set of 70% and 30%, respectively. Table 2 shows the sample distribution in the train and test splits.

Image preprocessing

The image preprocessing procedure consisted of two fundamental stages in our investigation: grayscale conversion and resizing. The process entails the transformation of the image to grayscale, adjusting its dimensions while ensuring the preservation of the aspect ratio, and resizing to dimensions of 224 pixels by 224 pixels.

ML

Due to the limited availability of images of a specific class within the provided data, it is feasible to achieve successful classification using conventional ML techniques. Therefore, if the classification task is successfully and effectively completed, switching to DL methodologies is unnecessary. However, if there is a need to do so, the ML methodology was implemented to ascertain the necessity of employing a DL approach. Eight distinct pipelines were generated in the ML approach, as illustrated in Table 3. Each pipeline incorporated a sequence of operations, including gray scaling, resizing, and standard scaling.

The input image is processed in the first pipeline by transforming it into a grayscale representation. Subsequently, LBP utilized a default method with a circle radius of 3 points and a circularly symmetric neighbor set of 8 × 3 points. This process resulted in the generation of a histogram. Subsequently, the data underwent standard scaling and achieved a zero mean and one standard deviation. Using LBP with the SVM classifier was helpful because LBP picked up on local differences in texture, which made it easier for the SVM classifier to articulate the difference between different patterns.

The second pipeline exhibited similarities to the first pipeline; however, it diverged in terms of the feature transformation process. The investigation employed PCA for dimension reduction subsequent to standard scaling. After performing PCA, the extracted components successfully captured most of the variance. Subsequently, an SVM with a Radial Basis Function (RBF) kernel was employed.

In the same way, the following two pipelines were similar to the first two, but instead of the default method of LBP, the uniform method was employed. Uniform LBP reduced the dimensionality of the feature space by mapping the non-uniform patterns to the uniform ones. The results for the eight pipelines of ML are shown in Table 4.

The prime difference between the last and first four pipelines was using weighted samples to tackle imbalanced sampling. In the first four pipelines, class weights for both classes were the same. Eq. (2), given the class weights for the following four pipelines, turned out to be around 0.52 for the negative class and about 12.40 for the positive class. Note that the weight given to the positive class is much higher since the number of samples is lower.

From Table 4, the following can be inferred:

- The first two was the poorest-performing pipelines, and
- The last two were the best-performing pipelines.

Table 4
Classification Report of ML Pipelines.

Pipeline	Accuracy	Specificity	Sensitivity	Precision	F1 Score
First	96.80	100.00	01.00	100.00	01.98
Second	96.80	100.00	01.00	100.00	01.98
Third	97.32	99.80	23.00	79.31	35.65
Fourth	97.29	99.80	22.00	78.57	34.37
Fifth	96.90	99.43	21.00	55.26	30.43
Sixth	96.74	99.33	19.00	48.71	27.34
Seventh	**94.32**	95.39	62.00	31.00	**41.34**
Eighth	94.09	95.13	63.00	30.14	40.78

Table 5
The Structure of Models Used in Transfer Learning.

Models	Dense 1	Dense 2	Dense 3	Trainable Parameters
First	1024	1024	512	2,100,738
Second	512	512	256	657,154
Third	256	256	128	230,274
Fourth	128	128	64	90,562
Fifth	1024	512	–	1,051,138
Sixth	512	256	–	394,498
Seventh	256	128	–	164,482
Eighth	128	64	–	74,050

The F1 scores of the first, second, fifth, and sixth pipelines exhibited lesser values compared to the third, fourth, seventh, and eighth pipelines. The LBP technique was utilized to extract microstructures to overcome the inherent limitations of SVM in directly classifying images.

PCA was employed for feature transformation after the extraction of features, as the performance of LBP in isolation was suboptimal. Additionally, it can be noted that using PCA to reduce dimensionality resulted in enhanced time and space efficiency, albeit at the expense of performance. However, despite attempts to improve performance, it was found that the default method yielded only a partial result. Consequently, the uniform method of LBP was employed as an alternative, leading to a notable improvement in performance. Ultimately, the implementation of weighted sampling yielded a noteworthy enhancement in performance. Weighted sampling was performed for the last four pipelines, as mentioned earlier, hoping to deal with imbalanced data, which improved the performance. However, it should be noted that the trained models did not meet the desired level of performance. It is evident from the F1 score of 41.34% and accuracy of 94.32% in the seventh model, which did not significantly improve in a binary classification scenario. Given the suboptimal performance of ML models, it appeared prudent to explore the application of CNN models as an alternative. Additionally, this method would reduce the need for subjective feature engineering.

Transfer learning with VGG16 and VGG19

Transfer learning on the ImageNet dataset using pre-trained VGG16 and VGG19 models has been applied because the quantity of samples is a few thousand. Two separate pipelines using the transfer learning methods VGG16 and VGG19 were built to make a diagnosis based on the provided data. There were a total of eight distinct models produced by each pipeline. The structure of these models is shown in Table 5.

The aim was to reduce the number of trainable parameters without hurting performance. A feature map was extracted from the convolution layers, which was used to train different models with varying dense layers, trying to achieve similar performance with a smaller number of learnable parameters.

The image labels underwent one-hot encoding. Subsequently, feature maps were acquired for the training and validation data using the pre-trained VGG16 and VGG19 models. The input shapes for these feature maps were (224, 224, 3), and the top layers were excluded from the process. The outputs derived from the pre-trained VGG16 and VGG19 models were in the shape of (7, 7, 512). Table 5 shows that these outputs went through variable dense layers with ReLU activation after a 2D layer that took the global average and pooled it. Finally, the outcomes were passed through an output-dense layer with two units and SoftMax activation.

The training process involved training each model for 20 and 50 epochs, utilizing a batch size 32. A categorical cross-entropy was used as a loss function, and Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.001 was used to train eight different models. Utilizing the SGD optimizer increases the probability of minimizing the error rate to its utmost extent. Throughout the learning process, the categorical cross-entropy loss function mainly calculated and minimized errors. Table 6 displays the hyperparameters that have been specified, including the optimizer, learning rate, loss function, batch size, and epochs.

The classification report for transfer learning using VGG16 and VGG19 is shown in Tables 7 and 8, respectively.

Based on the data presented in Tables 7 and 8, it is possible to deduce the following:

Table 6
Hyper-parameter Configuration.

Hyper-Parameter	Value
Optimizer	Stochastic Gradient Descent
Learning Rate	0.001
Loss Function	Categorical Cross-Entropy
Batch Size	32
Epochs	20 and 50

Table 7
Classification Report for Transfer Learning using VGG16.

Models	Epochs	Accuracy	Specificity	Sensitivity	Precision	F1 Score
First	20	98.32	99.93	50.00	96.15	65.79
	50	98.38	99.96	51.00	98.07	67.11
Second	20	98.16	99.43	60.00	77.92	67.80
	50	98.32	99.93	50.00	96.15	65.78
Third	20	97.32	99.83	22.00	81.48	34.65
	50	98.09	100.00	41.00	100.00	58.16
Fourth	20	97.38	99.83	24.00	82.75	37.21
	50	97.93	99.70	45.00	83.33	58.44
Fifth	20	98.41	99.80	56.99	90.47	69.93
	50	98.38	99.86	54.00	99.10	69.91
Sixth	20	98.00	99.33	57.99	74.35	65.16
	50	**98.45**	99.80	57.99	90.62	**70.96**
Seventh	20	97.51	99.63	34.00	75.55	46.89
	50	97.87	99.56	47.00	78.33	58.75
Eighth	20	97.70	99.80	35.50	85.36	50.15
	50	98.09	99.73	49.00	85.96	62.42

Table 8
Classification Report for Transfer Learning using VGG19.

Models	Epochs	Accuracy	Specificity	Sensitivity	Precision	F1 Score
First	20	97.54	99.96	25.00	96.15	39.69
	50	97.87	99.86	38.00	90.47	53.53
Second	20	97.64	99.90	30.00	90.90	45.12
	50	98.00	99.66	48.00	82.75	60.76
Third	20	96.83	99.46	18.00	52.94	26.87
	50	98.12	99.73	50.00	86.20	63.29
Fourth	20	97.25	99.56	28.00	68.29	39.72
	50	97.64	99.80	33.00	84.61	47.49
Fifth	20	97.51	99.80	28.99	82.85	42.95
	50	98.35	99.86	53.00	99.98	69.28
Sixth	20	97.83	99.73	41.00	83.67	55.04
	50	97.83	99.60	45.00	78.94	57.33
Seventh	20	97.38	99.70	28.00	75.67	40.88
	50	98.29	99.96	48.00	97.95	64.43
Eighth	20	97.09	99.80	16.00	72.72	26.23
	50	97.70	99.40	47.00	72.30	56.97

- Within the context of ML pipelines, it was observed that the model exhibiting the highest level of performance achieved an F1 score of 41.34%. It is worth noting that, despite the accuracy metric yielding a value of 94.32%, this measure proved deceptive. Transfer learning has been shown to outperform ML methods. In the sixth model, which utilized the VGG16 architecture, transfer learning achieved an F1 score of 70.96% and an accuracy of 98.45%.
- The number of trainable parameters could be reduced without significantly hurting performance.
- The performance of models trained for 50 epochs surpasses that of models trained for 20 epochs. In the context of the employed dataset, it has been observed that VGG16 can extract a more optimal feature map than VGG19.
- Utilizing two dense layers instead of three improved performances, even though the number of trainable parameters had been reduced.
- The performance of VGG19 was comparatively inferior to that of VGG16.

The examination of transfer learning using VGG16 and VGG19 discussed did not entail using class weight. In the subsequent phase, the weighting scheme for classes was modified to address the issue of imbalanced datasets, while the gradient clipping threshold was established at 0.5 to mitigate the problem of exploding gradients. Eq. (2) provided the values for the class weights. The eight models in Table 5 underwent retraining using VGG-16 and VGG-19 architectures. The results of this retraining are depicted in Tables 9 and 10.

Table 9
Classification Report with Weighted Samples for Transfer Learning Using VGG16.

Models	Epochs	Accuracy	Specificity	Sensitivity	Precision	F1 Score
First	20	98.48	99.10	80.00	74.76	77.30
	50	98.90	99.73	74.00	88.26	80.64
Second	20	98.48	99.23	76.00	76.76	76.38
	50	98.41	99.40	69.00	79.31	73.80
Third	20	97.74	98.46	76.00	62.29	68.47
	50	98.67	99.40	77.00	81.05	78.98
Fourth	20	97.61	98.06	84.00	59.15	69.42
	50	98.16	99.00	73.00	70.87	71.92
Fifth	20	98.51	99.40	72.00	80.00	75.79
	50	**98.83**	99.53	78.00	84.78	**81.25**
Sixth	20	98.35	99.03	78.00	72.89	75.36
	50	98.64	99.40	76.00	80.85	78.38
Seventh	20	98.58	99.40	74.00	80.43	77.09
	50	98.38	99.00	80.00	72.72	76.19
Eighth	20	96.67	96.96	88.00	49.16	63.09
	50	98.29	99.16	72.00	74.22	73.10

Table 10
Classification Report with Weighted Samples for Transfer Learning Using VGG19.

Models	Epochs	Accuracy	Specificity	Sensitivity	Precision	F1 Score
First	20	97.51	98.13	79.00	58.51	67.23
	50	98.45	99.13	78.00	75.00	76.48
Second	20	96.19	96.73	80.00	44.94	57.56
	50	97.96	98.73	75.00	66.37	70.43
Third	20	97.70	98.40	77.00	61.60	68.45
	50	98.22	98.80	81.00	69.23	74.66
Fourth	20	93.80	94.19	82.00	32.03	46.07
	50	97.83	98.63	74.00	64.34	68.84
Fifth	20	98.09	98.86	75.00	68.80	71.77
	50	98.48	99.16	78.00	75.72	76.85
Sixth	20	95.80	96.06	88.00	42.71	57.51
	50	98.32	99.10	75.00	73.52	74.26
Seventh	20	96.74	97.39	77.00	49.67	60.39
	50	97.74	98.56	73.00	62.93	67.60
Eighth	20	96.96	97.96	67.00	52.34	58.77
	50	97.38	98.03	78.00	56.93	65.82

Upon examination of Tables 9 and 10, it is possible to deduce the following:

- It is evident that the Fifth model, utilizing the VGG16 architecture, exhibited superior performance. Specifically, it achieved an F1 score of 81.25% and an accuracy of 98.83% following 50 epochs.
- In contrast to the previously reported findings on transfer learning with VGG16, which yielded an F1 score of 70.96% and an accuracy of 98.45%, the new investigation observed a better outcome.
- In the context of transfer learning with VGG16 with weighted samples, the fifth model achieved an F1 score of 81.25%, making it the highest-performing model. The first model also demonstrated an accuracy of 98.90% but a lower F1 score, which was misleading, indicating its reliability as it exhibits minimal misclassification in both classes.
- The performance of the VGG19 model demonstrated a relatively lower level of effectiveness when compared to the VGG16 model when weighted sampling was employed.
- The finding illustrates that using weighted sampling can potentially enhance a model's ability to handle imbalanced datasets effectively.

Weighted sampling was employed to address the challenges associated with imbalanced data, thereby contributing to enhanced performance, as previously elucidated. Furthermore, it is essential to consider the potential impact of the aforementioned proposed models on their overall effectiveness. Therefore, we employed sequential CNN architectures with and without weighted samples. The methods that were suggested made use of the criteria mentioned earlier to train in sequential order and for evaluation as well.

Evaluation of the proposed methods

Proposed methods

The images were subjected to preprocessing, wherein the labels were encoded using the one-hot encoding technique. Additionally, the hyperparameters were applied, as mentioned earlier. The selection of multiple filters in each convolutional layer enabled extracting a wide range of features at various scales. By selecting the most valuable activations, max pooling preserved the strongest ones,

Table 11
Classification Report for Proposed Models.

Models	Epochs	Accuracy	Specificity	Sensitivity	Precision	F1 Score
First	20	97.74	99.93	32.00	94.11	47.77
	50	97.74	99.86	34.00	89.47	49.27
Second	20	**97.67**	100.00	28.00	100.00	43.75
	50	97.64	99.90	30.00	90.90	45.12

Table 12
Confusion Matrix for Proposed Models.

Models	Epochs	True Negative	False Positive	False Negative	True Positive
First	20	2998	2	68	32
	50	2996	4	66	34
Second	20	3000	0	72	28
	50	2997	3	70	30

Table 13
Classification Report for Proposed Models with Weighted Sampling.

Models	Epochs	Accuracy	Specificity	Sensitivity	Precision	F1 Score
First	20	99.35	99.86	84.00	95.45	89.35
	50	**99.45**	99.90	86.00	96.62	**91.00**
Second	20	99.03	99.90	73.00	96.05	82.95
	50	99.00	99.70	78.00	89.65	83.43

Table 14
Confusion Matrix for Proposed Models with Weighted Sampling.

Models	Epochs	True Negative	False Positive	False Negative	True Positive
First	20	2996	4	16	84
	50	2997	3	14	86
Second	20	2997	3	27	73
	50	2991	9	22	78

contributing to robust feature identification. The flattening layer was intermediate between the convoluted and dense layers and transformed the three-dimensional feature maps into a one-dimensional vector. It enabled the dense layers to classify data using features that had been learned. The dense layers with densely interconnected neurons analyzed the extracted features and identified patterns indicative of particular classes. The gradual reduction in the number of neuron units in the dense layers (1024, 1024, 512) enabled a hierarchical representation of features, which promoted a more efficient learning process. Effectively introducing non-linearity into the network, the ReLu activation function enabled the modeling of intricate relationships between features. By discarding negative values, ReLu reduced the likelihood of gradients disappearing and accelerated training convergence. The Softmax activation function was used for the classification, generating a probability distribution over the output classes. The confusion matrix comprehensively represents the total count of instances that constitute the test dataset. The classification report, presented in Table 11, and the confusion matrix, depicted in Table 12, are provided below.

After analyzing Table 11, it is feasible to infer the following:

- The second proposed model yielded a result of 97.67% accuracy, with specificity and precision values of 100.00% and 100.00%, respectively, at 20 epochs. The F1 score attained a value of 86.45%, falling short of the desired threshold that necessitated the utilization of weighted samples.
- Additionally, it is noteworthy that the first model achieved an accuracy rate of 97.74%. The specificity and precision values are 98.93% and 94.11%, respectively. Furthermore, the F1 score reached a value of 49.27% after 50 epochs, which is not close to the performance of a particularly effective model..

Proposed methods with weighted sampling

The utilization of weighted sampling results in the presentation of the classification report and the confusion matrix are displayed in Tables 13 and 14, respectively. The confusion matrix comprehensively represents the total count of instances that constitute the test dataset.

After analyzing Table 13, it is feasible to infer the following:

- The first proposed model demonstrated that using weighted sampling provided an F1-score of 91.00%, an accuracy of 99.45%, and a specificity of 99.90% at 50 epochs, the highest possible score based on all of the investigations.

Table 15
Comparative analysis.

Author	(Dataset) – X-rays Samples	CNN Method	Accuracy (%)
Elene Firmeza Ohata et al. [14]	(Dataset-A) - COVID-19 - 194 Healthy - 194	Transfer Learning (MobileNet + SVM)	98.46
	(Dataset-B) - COVID-19 - 194 Healthy - 194	Transfer Learning (Densenet + MLP)	95.64
Ayturk Keles et al. [15]	COVID-19 - 210 Normal - 350	COV19-CNNet (CNN + 4 Dense Layer)	94.28
		COV19-ResNet (ResNet)	97.61
K. Hassan et al. [35]	(Custom Dataset) - Normal – 1400 COVID-19 - 100 (COVIDx) - Normal – 8066 COVID-19 - 183	Faster R–CNN (VGG-16 Network-based Faster Regions + CNN)	97.36
Vishu Madaan et al. [40]	COVID-19+ – 196 COVID-19 - – 1583 Other - 5027	XCOVNet (Sequential CNN)	98.44
K. Shaheed et al. [41]	COVID-19 - 375 Normal - 375	CNN+ Random Forest	97.00
This Research	(COVID-Xray-5k) - COVID-19+ – 184 COVID-19 - – 5000	Sequential CNN	99.45

- Incorporating weighted sampling into the proposed models, in conjunction with the assessment as mentioned earlier criteria, yielded better results.

Comparative analysis

This research achieved a significant F1-score of 91.00%, which indicated the method's effectiveness in accurately identifying positive and negative instances. Additionally, it demonstrated improved classification performance, as evidenced by having the highest accuracy rate of 99.45% among all the applied methods. The comparative analysis was performed by contrasting the results of the present investigation with the previous research, as illustrated in Table 15.

The COVID-Xray-5k dataset employed in the investigation comprised many samples categorized as positive for COVID-19 or negative for COVID-19. While specific studies incorporated transfer learning or a combination of CNN and other techniques, this research employed ML and transfer learning using VGG16 and VGG19 models, both with and without weighted sampling. Additionally, the research proposes a sequential CNN architecture. The proposed sequential CNN architecture facilitated the process of implementation and interpretation for better understanding. According to the findings, this research outperformed the others regarding F1-score and accuracy, used a large dataset, and applied a relatively simple procedure.

Conclusions

The research aimed to create an image classification model capable of adequately predicting X-ray images that fall into one of two major classes with a reasonable degree of effectiveness, which was achieved. The investigation indicates that the ML models' performance is below the expected standard. Specifically, the models achieved an F1 score of 41.34% and an accuracy of 94.32%. Consequently, attention is redirected toward employing transfer learning techniques using the VGG16 and VGG19 models. These models yielded significantly improved results with weighted sampling, with an F1 score of 81.25% and an accuracy of 98.83%. The first proposed method employed a weighted sample and produced encouraging results, with an F1-score of 91.00% and an accuracy of 99.45%, respectively. The experiment results show that the recommended method of using sequential CNN and transfer learning with VGG16 and VGG19 models can be used for classification tasks.

Furthermore, the methods mentioned earlier utilizing weighted sampling have demonstrated that it is possible to attain a satisfactory level of performance by reducing the number of trainable parameters in the model. The COVID-Xray-5k dataset exhibited an imbalance characterized by a notable discrepancy in the distribution of X-ray images. This imbalance could have affected how well the detection system worked, which could have led to a bias in favor of the majority class. An investigation is also needed to determine how well the results can be applied to various datasets and real-world situations. Dataset imbalances must be considered to improve system resilience. This method could enhance the model's learning from both classes while mitigating the risk of potential biases. Even though the results are very promising, there is still scope for further research on a larger dataset and modern ML techniques, such as new deep learning architectures or ensemble models, to identify potential improvement areas.

Ethics statements

The publicly available dataset is available for investigation by anybody, and no permission is required, as the owner of the dataset generator already authorized it.

Declaration of Competing Interest

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

CRediT authorship contribution statement

Sunil Kumar: Conceptualization, Methodology, Data curation, Writing – original draft. **Harish Kumar:** Supervision, Validation, Writing – review & editing.

Data availability

The authors do not have permission to share data.

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] WHO director-general's opening remarks at the media briefing on COVID-19 - 11 march 2020, World Health Organization. (n.d.). https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19-11-march-2020 (accessed December 5, 2022).

[2] D.K. Redie, A.E. Sirko, T.M. Demissie, S.S. Teferi, V.K. Shrivastava, O.P. Verma, et al., Diagnosis of COVID-19 using chest X-ray images based on modified darkcovidnet model, Evol Intell (2022), doi:10.1007/s12065-021-00679-7.

[3] A.K. Das, S. Ghosh, S. Thunder, R. Dutta, S. Agarwal, A. Chakrabarti, Automatic covid-19 detection from X-ray images using ensemble learning with convolutional neural network, Pattern Anal Appl 24 (2021) 1111–1124, doi:10.1007/s10044-021-00970-4.

[4] S. Minaee, R. Kafieh, M. Sonka, S. Yazdani, G.Jamalipour Soufi, Deep-COVID: predicting COVID-19 from chest X-ray images using deep transfer learning, Med Image Anal 65 (2020) 101794, doi:10.1016/j.media.2020.101794.

[5] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, R.M. Summers, Chestx-Ray8: hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases, in: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, doi:10.1109/cvpr.2017.369.

[6] V. Kumawat, B. Umamaheswari, P. Mitra, G. Lavania, Machine learning for health care: challenges, controversies, and its applications, Soft Comput (2022) 253–261, doi:10.1007/978-981-19-0707-4_24.

[7] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans Pattern Anal Mach Intell 24 (2002) 971–987, doi:10.1109/tpami.2002.1017623.

[8] S. Garg, G. Hamarneh, A. Jongman, J.A. Sereno, Y. Wang, ADFAC: automatic detection of facial articulatory features, MethodsX 7 (2020) 101006, doi:10.1016/j.mex.2020.101006.

[9] G. Marchioro, C. Daffara, PCA-based method for managing and analyzing single-spot analysis referenced to spectral imaging for artworks diagnostics, MethodsX 7 (2020) 100799, doi:10.1016/j.mex.2020.100799.

[10] N. Lavanya Devi, P. Thirumurugan, Cervical Cancer Classification from PAP SMEAR images using modified fuzzy c means, PCA, and KNN, IETE J Res 68 (2021) 1591–1598, doi:10.1080/03772063.2021.1997353.

[11] C. Chen, H. Seo, C.H. Jun, Y. Zhao, Pavement crack detection and classification based on fusion feature of LBP and PCA with SVM, Int J Pavement Eng 23 (2021) 3274–3283, doi:10.1080/10298436.2021.1888092.

[12] C.E. da S. Santos, L. dos S. Coelho, C.H. Llanos, Nature inspired optimization tools for SVMs - NIOTS, MethodsX 8 (2021) 101574, doi:10.1016/j.mex.2021.101574.

[13] E. Hussain, M. Hasan, M.A. Rahman, I. Lee, T. Tamanna, M.Z. Parvez, Corodet: a deep learning based classification for COVID-19 detection using chest X-ray images, Chaos, Solitons Fractals 142 (2021) 110495, doi:10.1016/j.chaos.2020.110495.

[14] E.F. Ohata, et al., Automatic detection of COVID-19 infection using chest X-ray images through transfer learning, IEEE/CAA J Autom Sin 8 (1) (2021) 239–248, doi:10.1109/JAS.2020.1003393.

[15] A. Keles, M.B. Keles, A. Keles, COV19-CNNet and COV19-ResNet: diagnostic inference engines for early detection of COVID-19, Cognit Comput (0123456789) (2021), doi:10.1007/s12559-020-09795-5.

[16] L. Kong, J. Cheng, Classification and detection of COVID-19 X-Ray images based on DenseNet and VGG16 feature fusion, Biomed Signal Process Control 77 (2022) 103772, doi:10.1016/j.bspc.2022.103772.

[17] R.M.A. Ikram, R.R. Mostafa, Z. Chen, K.S. Parmar, O. Kisi, M. Zounemat-Kermani, Water temperature prediction using improved deep learning methods through reptile search algorithm and weighted mean of vectors optimizer, J Mar Sci Eng 11 (2023) 259, doi:10.3390/jmse11020259.

[18] M.O. Alassafi, M. Jarrah, R. Alotaibi, Time series predicting of COVID-19 based on deep learning, Neurocomputing 468 (2022) 335–344, doi:10.1016/j.neucom.2021.10.035.

[19] E.K. Juarez, M.R. Petersen, A comparison of machine learning methods to forecast tropospheric ozone levels in Delhi, Atmosphere (Basel) 13 (2021) 46, doi:10.3390/atmos13010046.

[20] Z. Zhang, R. Yang, Y. Fang, LSTM network based on antlion optimization and its application in flight trajectory prediction, in: Proceedings of the 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2018, doi:10.1109/imcec.2018.8469476.

[21] P. Kora, C.P. Ooi, O. Faust, U. Raghavendra, A. Gudigar, W.Y. Chan, K. Meenakshi, K. Swaraja, P. Plawiak, U.Rajendra Acharya, Transfer learning techniques for medical image analysis: a review, BBE 42 (2022) 79–107, doi:10.1016/j.bbe.2021.11.004.

[22] A. Narin, C. Kaya, Z. Pamuk, Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks, Pattern Anal Appl 24 (2021) 1207–1220, doi:10.1007/s10044-021-00984-y.

[23] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint, (2014). arXiv:1409.1556.

[24] M. Abdel-Basset, N.N. Mostafa, K.M. Sallam, I. Elgendi, K. Munasinghe, Enhanced COVID-19 X-ray image preprocessing schema using type-2 neutrosophic set, Appl Soft Comput 123 (2022) 108948, doi:10.1016/j.asoc.2022.108948.

[25] W.K. Mutlag, S.K. Ali, Z.M. Aydam, B.H. Taher, Feature extraction methods: a review, J Phys Conf Ser 1591 (2020) 012028, doi:10.1088/1742-6596/1591/1/012028.

[26] A. Alqudah, A.M. Alqudah, Sliding window based support vector machine system for classification of breast cancer using histopathological microscopic images, IETE J Res 68 (2019) 59–67, doi:10.1080/03772063.2019.1583610.

[27] H. Verma, A. Vidyarthi, A.V. Chitre, K.H. Wanjale, M. Anusha, A. Majrashi, S.K. Hinga, Local binary patterns based on neighbor-center difference image for color texture classification with machine learning techniques, Wireless Commun Mobile Comput 2022 (2022) 1–11, doi:10.1155/2022/1191492.

[28] G. King, Z. Langche, Logistic regression in rare events data, Political Anal 9 (2001) 137–163 https://gking.harvard.edu/files/0s.pdf.

[29] A.F.M. Alkarkhi, Z-test for one-sample mean, Applications of Hypothesis Testing for Environmental Science. (2021) 13–32. doi:10.1016/b978-0-12-824301-5.00007-1.

[30] W. Zhang, Y. Du, T. Yoshida, Y. Yang, DeepRec: a deep neural network approach to recommendation with item embedding and weighted loss function, Information Sciences, Elsevier Science. 470 (2019) 121–140, ISSN 0020-0255. https://in.booksc.eu/book/72118984/f89d33.

[31] D. Sharifrazi, R. Alizadehsani, M. Roshanzamir, J.H. Joloudari, A. Shoeibi, M. Jafari, et al., Fusion of convolution neural network, support vector machine and Sobel filter for accurate detection of COVID-19 patients using X-ray images, Biomed Signal Process Control 68 (2021) 102622, doi:10.1016/j.bspc.2021.102622.

[32] C.R. Kishore, R. Pemula, S. Vijaya Kumar, K.P. Rao, S. Chandra Sekhar, Deep learning models for identification of COVID-19 using CT images, Soft Comput (2022) 577–588, doi:10.1007/978-981-19-0707-4_52.

[33] T. Garg, M. Garg, O.P. Mahela, A.R. Garg, Convolutional Neural Networks with transfer learning for recognition of COVID-19: a comparative study of different approaches, AI 1 (2020) 586–606, doi:10.3390/ai1040034.

[34] L. Wang, Z.Q. Lin, A. Wong, COVID-Net: a tailored deep convolutional neural network design for detection of COVID19 cases from chest X-ray images, Sci Rep 10 (2020) 19549, doi:10.1038/s41598-020-76550-z.

[35] K. Hassan, S. Kumar, T. Islam, M. Rahman, Informatics in medicine unlocked COVID faster R –CNN: a novel framework to diagnose novel coronavirus disease (COVID-19) in X-Ray images, Informatics Med Unlocked 20 (2020) 100405, doi:10.1016/j.imu.2020.100405.

[36] A.M. Lad, A. Mishra, A. Rajagopalan, Comparative analysis of convolutional neural network architectures for real time COVID-19 facial mask detection, J Phys Conf Ser 1969 (2021) 012037, doi:10.1088/1742-6596/1969/1/012037.

[37] J. Dessain, Machine learning models predicting returns: why most popular performance metrics are misleading and proposal for an efficient metric, Expert Syst Appl 199 (2022) 116970, doi:10.1016/j.eswa.2022.116970.

[38] R.E. Al Mamlook, S. Chen, H.F. Bzizi, Investigation of the performance of machine learning classifiers for pneumonia detection in chest X-ray images, in: Proceedings of the 2020 IEEE International Conference on Electro Information Technology (EIT), 2020, doi:10.1109/eit48999.2020.9208232.

[39] C.E. Metz, Basic principles of ROC analysis, Semin Nucl Med 8 (4) (1978) 283–298.

[40] V. Madaan, et al., XCOVNet: chest x-ray image classification for COVID-19 early detection using convolutional neural networks, New Gener Comput (0123456789) (2021), doi:10.1007/s00354-021-00121-7.

[41] K. Shaheed, P. Szczuko, Q. Abbas, A. Hussain, M. Albathan, Computer-aided diagnosis of COVID-19 from chest x-ray images using hybrid-features and random forest classifier, Healthcare 11 (2023) 837, doi:10.3390/healthcare11060837.