COVID-MRSNet: A Novel Deep Convolutional Neural Network Architecture for Automatic classification of coronavirus (SARS-CoV-2), pneumonia and healthy lungs from CXR images

CHAIMAE OUCHICHA
Department of Management Assistance Techniques,
Hassan 1st University (UH1), ENCG, FAMISDS
Lab, Settat,
MOROCCO

Abstract: - December 2019 marked the emergence of a novel coronavirus in China that rapidly spread worldwide. Early detection of COVID-19 is crucial to limit transmission. The standard diagnostic method, reverse transcription-polymerase chain reaction (RT-PCR), is accurate but costly and time-consuming. Recent studies indicate that deep learning models can reliably detect COVID-19 from chest X-ray (CXR) images, offering a faster and more cost-effective alternative. In this paper, we propose COVID-MRSNet, a deep Convolutional Neural Network-based ResNet architecture for classifying COVID-19, viral pneumonia, and normal cases using CXR images. The model was trained and evaluated on a public dataset of 2,905 images, including 219 COVID-19, 1,345 viral pneumonia, and 1,341 normal cases. After 50 epochs, COVID-MRSNet achieved an average accuracy of 98.34%, recall of 98.67%, precision of 98.55%, and F1-score of 98.61%. The results demonstrate superior multi-class classification performance compared to existing CNN-based methods.

Key-Words: - Artificial intelligence, deep learning, Convolutional neural models, Chest radiography, COVID-19 detection, Pulmonary disease classification, Medical image analysis.

Received: May 19, 2025. Revised: September 8, 2025. Accepted: December 9, 2025. Published: May 5, 2026.

1 Introduction

Since its first appearance in Wuhan, China in late 2019, COVID-19 has rapidly become a global health crisis, affecting populations across virtually every country and posing significant challenges to healthcare systems. This highly contagious disease has now been reported in nearly all countries and is associated with severe health complications. The early diagnosis of COVID-19 is crucial not only for improving patient treatment outcomes but also for protecting public health through effective isolation strategies and epidemic containment.

Real-time reverse transcription polymerase chain reaction, [1], has been widely adopted as the reference standard for COVID-19 diagnosis. However, this technique suffers from several limitations, including its manual nature, high cost, complex processing requirements, long turnaround time, and relatively low sensitivity ranging between 60–70%. In contrast, medical imaging modalities play a critical role in assessing respiratory diseases. In particular, radiographic imaging has been shown to exhibit higher sensitivity than conventional PCR-based testing methods, [2], [3]. Additionally, chest X-ray imaging is more affordable and widely accessible than computed tomography (CT) systems due to its faster acquisition process, lower radiation exposure, and availability in most healthcare facilities, [4]. As a result, clinicians increasingly rely on chest CT and chest X-ray findings to support diagnostic decisions, ([5], [6]).

Chest X-ray images are especially valuable in COVID-19 diagnosis, as they reveal characteristic patterns such as lung opacity and diffuse infiltrates. These visual features provide sufficient information to determine infection status and enable early disease detection. This diagnostic approach has been particularly important in regions facing shortages of RT-PCR testing kits, prompting healthcare professionals to adopt radiographic imaging as a practical alternative.

Advances in medical imaging acquisition technologies have facilitated the large-scale collection of radiological data, making extensive chest X-ray datasets readily available. Nevertheless, manual interpretation and classification of COVID-19 chest X-ray images remain challenging and time-consuming tasks for radiologists. Consequently, there is a growing need for automated diagnostic systems that can reduce diagnostic workload, minimize human error, and alleviate clinician fatigue.To meet this requirement, computer-aided diagnostic tools have been created to help extract important clinical information from medical imaging data.

Artificial Intelligence (AI) has emerged as one of the most influential technologies globally, contributing significantly to monitoring the spread of the coronavirus, analyzing infection growth rates, and assessing patient risk levels and disease severity. Furthermore, AI-based analytical models can estimate mortality risk by leveraging historical patient data, thereby aiding clinicians in making informed

decisions. Machine learning and deep learning approaches have become widely applied in medical image analysis for detecting and classifying diseases. Unlike traditional diagnostic methods, AI-based systems can automatically extract discriminative features from medical images, improving diagnostic performance, assisting radiologists, and enhancing patient care. Among these approaches, deep learning models based CNNs have proven highly effective in various medical imaging applications and are now a fundamental element of modern image analysis workflows.

In this study, an innovative deep learning approach, referred to as COVID-MRSNet, is introduced for automatically identify COVID-19 from chest X-ray scans. The developed framework is built upon a ResNet-inspired architecture, which is employed both for extracting discriminative visual features and for performing disease classification.

COVID-MRSNet is developed and evaluated on a publicly available dataset consisting of 2,905 chest X-ray images, including 219 COVID-19 cases, 1,341 normal instances, and 1,345 pneumonia samples (both bacterial and viral). To ensure a robust and unbiased assessment, model performance is measured using k-fold cross-validation. The evaluation employs several standard metrics such as accuracy, precision, sensitivity, specificity, F1-score, alongside confusion matrices, ROC curves, and the area under the ROC curve (AUC).

Results from the test sets demonstrate that COVID-MRSNet achieves highly competitive performance in multi-class classification tasks, effectively distinguishing between COVID-19, pneumonia, and healthy cases. When compared to multiple leading CNN-based approaches, COVID-MRSNet consistently shows superior accuracy, robustness, and reliability for COVID-19 detection using chest X-ray images.

The key contributions of this work are highlighted as follows:

1. We introduce COVID-MRSNet, a novel deep learning framework inspired by ResNet, designed to automatically identify key patterns and features from chest X-ray images, enabling accurate classification of different lung conditions. A thorough comparison with existing state-of-the-art classifiers is conducted

2. We present a novel MRC block that efficiently performs downsampling while simultaneously expanding the number of feature channels, improving both computational efficiency and feature representation.

3. The proposed framework is trained and assessed using publicly accessible datasets under two experimental protocols: a 5-fold cross-validation scheme and a hold-out validation approach.

4. COVID-MRSNet demonstrates excellent performance, with accuracy reaching 98.34%, recall reported as 98.67%, precision measured at 98.55%, and an F1-score attaining 98.55%.

5. The classification performance of COVID-MRSNet surpasses that of previously reported methods cited in this study.

6. The proposed approach enables COVID-19 detection without the need for feature selection techniques or manual feature extraction.

The structure of the paper is outlined as follows. Section 2 presents a review of the related literature. In Section 3, the proposed COVID-MRSNet architecture is described in detail. Section 4 provides information about the datasets used in this study, as well as the training procedure and evaluation criteria. Section 5 presents the outcomes of the experiments along with their interpretation. In Section 6, we provide a final discussion, highlighting the key insights of this study and suggesting potential avenues for further investigation.

2 State-of-the-Art Methods

The World Health Organization has confirmed that COVID-19 has rapidly spread across numerous countries, posing a serious global health challenge. Early identification of infected individuals is crucial for controlling viral transmission and reducing the burden on healthcare systems. In response to this urgent need, extensive research efforts have been directed toward developing efficient diagnostic tools. Among these, chest X-ray (CXR) imaging has emerged as a widely used and cost-effective modality for screening respiratory infections. However, the manual interpretation of CXR images is time-consuming and subject to inter-observer variability, which has motivated the adoption of artificial intelligence (AI)-driven solutions.

In recent years, approaches based on deep learning, especially CNNs, have demonstrated strong potential for automating COVID-19 detection from medical imaging. Such networks can extract distinctive features directly from raw images, resulting in high diagnostic performance. Consequently, the use of CNNs for automated COVID-19 analysis of chest X-ray (CXR) images has become a major focus of research, with these architectures widely acknowledged as highly effective in this area.

DarkNet, presented in [7], is a convolutional neural network (CNN) framework developed for automated COVID-19 detection using chest X-ray images sourced from two publicly available datasets. The network comprises 17 convolutional layers that utilize a variety of filtering techniques. Its performance was assessed on both binary classification tasks (COVID-19 vs. no findings) and multi-class classification tasks (COVID-19 vs. viral pneumonia vs. no findings), achieving accuracies of 98.08% and 87.02%, respectively.

COVID-Net, introduced in [8], is a specialized deep CNN architecture tailored for detecting COVID-19 from chest X-ray images. The architecture incorporates a lightweight projection–expansion–projection–extension (PEPX) module to improve feature extraction while keeping computational costs low. Using the COVIDx dataset, which contains 13,975 X-ray images from various sources, the model reached a COVID-19 sensitivity of 91.0%.

DeepCoroNet, presented in [9], is a deep learning-based framework developed for automated COVID-19 detection using X-ray images. The model was evaluated on datasets comprising COVID-19, pneumonia, and normal cases, and demonstrated strong performance in three-class classification scenarios, outperforming several existing approaches.

CVR-Net was introduced in [10], as an end-to-end deep learning model capable of analyzing both CT and X-ray images. The framework integrates multi-scale and multi-encoder feature extraction mechanisms to capture complementary information from different representations. Extensive experiments conducted on multiple public datasets showed that CVR-Net achieved high accuracy and F1-score values across a variety of binary and multi-class classification tasks.

In [11], a deep learning framework for COVID-19 classification was developed, leveraging established neural network architectures such as VGG16, [12], ResNet, [13], and deep convolutional neural networks (DCCNs), [14]. The model was trained on an extensive CT dataset compiled from multiple sources. When compared to fifteen contemporary approaches, this framework consistently outperformed them in terms of specificity, sensitivity, AUC, F-measure, and overall accuracy.

CoroNet, described in [15], builds upon the Xception architecture pre-trained on ImageNet to classify COVID-19 cases from chest X-ray images. Training was conducted using images from two publicly available datasets. Evaluation results indicated that the model reached 95% accuracy for the three-class task, while the four-class task achieved 89.6% overall accuracy, supported by strong precision and recall measures.

A capsule network-based model, CapsNet, was investigated in [16], for detecting COVID-19 from CXR images. The model was applied to both binary and multi-class classification scenarios. It achieved nearly 97.24% accuracy for binary classification and 84.22% for multi-class tasks, demonstrating robust performance across different evaluation settings.

The study in [17], explored a multi-CNN framework that combines features extracted from several pre-trained CNN models. Correlation-based feature selection and a BayesNet classifier were employed to enhance classification performance. Experimental results on two public datasets showed high accuracy and AUC values, demonstrating the effectiveness of feature fusion strategies.

A patch-based CNN methodology was presented in [18], where random patch extraction and majority voting were used to determine final predictions. The framework integrates FC-DensNet103 for segmentation-based biomarker extraction and ResNet-18 for classification, achieving strong results on multi-class chest X-ray datasets.

COVNet, proposed in [19], is a three-dimensional deep learning framework that utilizes ResNet-50 as its backbone network. The model extracts both local and global features from chest CT scans to distinguish COVID-19 from other pulmonary diseases, achieving high sensitivity, specificity, and AUC values.

In [20], the authors introduced COVID-CAPS, a capsule network-based architecture incorporating convolutional layers and a modified loss function to address data imbalance. The model demonstrated strong performance in terms of accuracy, sensitivity, specificity, and AUC.

COVIDX-Net, presented in [21], consists of seven CNN architectures, including VGG19, DenseNet, and MobileNetV2. Each network analyzes normalized CXR images to classify COVID-19 cases. Among the evaluated models, VGG19 and DenseNet achieved the highest F1-scores.

CVDNet was developed in [22], as a deep architecture featuring parallel convolutional layers with varying kernel sizes to capture both local and global image features. The model achieved high accuracy for both binary and three-class COVID-19 classification tasks.

A Faster R-CNN-based diagnostic framework using the VGG-16 backbone was developed in [23]. The reported results indicate that this method was effective in identifying COVID-19 related patterns in chast X-ray images.

CoroDet, introduced in [24], is a deep CNN model capable of analyzing both CXR and CT images for COVID-19 detection. The framework supports binary, three-class, and four-class classification tasks

and demonstrated consistent performance across all configurations.

Finally, in [25], the authors proposed a CNN-based classification framework for distinguishing COVID-19, pneumonia, and normal cases using a large-scale dataset of X-ray and CT images. The model achieved high accuracy, sensitivity, and specificity across training, validation, and testing phases. Additional studies have further confirmed the effectiveness of deep CNN models for COVID-19 diagnosis using medical imaging modalities, [26], [27], [28].

3 The proposed COVID-MRSNet model

In this section, we introduce the COVID-MRSNet architecture developed for COVID-19 detection using chest X-ray images. The network is structured around seven essential components: the MRC block (Figure 1) (Appendix), convolutional layers, pooling layers, activation functions, batch normalization, dropout, and fully connected (dense) layers. Each of these elements is discussed in detail below, and the complete architecture of the proposed CNN for COVID-19 classification is presented in Figure 2 (Appendix).

(A) Convolutional layer

This is the layer where the convolution operation occurs, and the convolutional neural network model learns. Convolution is a major operation of the CNN, which allows features to be automatically extracted.

The convolution operation consists of taking a filter of a given size and dragging it over an image, calculating at each position the sum of the scalar product between the value of the inputs and the associated weights of the filter. Applying the same weights everywhere on the image helps instill in the network a prior that similar visual features can be present in multiple locations in the image. This is an advantage for convolution because far fewer parameters are required, both reducing the amount of memory and decreasing the number of operations that need to be performed. In addition, this operation has a smaller receptive field compared to that of a fully connected layer. It is also important to note that the deeper a layer is, the greater its receptive field on the input data. We can assume that this is one of the reasons why the network learns to extract hierarchical features.

In a convolutional neural network, each feature map in a convolutional layer is generated by applying a set of learnable filters to the feature maps of the preceding layer. This process involves performing a convolution operation between the input feature maps and their corresponding kernels, followed by the addition of a bias term. The resulting linear response is then transformed using a non-linear activation function in order to enhance the representational capacity of the network. This operation can be mathematically expressed as follows in Eq. 1:

$$C_j^l = \vartheta\left(\sum_{i \in N_i} I_i^{l-1} \otimes w_{i,j}^l + b_j^l\right) \tag{1}$$

Here, $\vartheta(\cdot)$ represents the activation function, and $\otimes$ denotes the convolution operation. The symbol I_i^{l-1} refers to the i^{th} feature map from the $(l-1)^{\text{th}}$ layer, whereas C_j^l indicates the j^{th} feature map generated in the l^{th} layer. The set N_i comprises all input feature maps that are connected to the current convolutional unit. The weight parameter $w_{i,j}^l$ defines the convolutional kernel linking the i^{th} input feature map to the j^{th} output feature map, and b_j^l is the bias term associated with the j^{th} output feature map in layer l.

A convolutional layer is defined by four principal hyperparameters that determine its functionality and learning capability. The first hyperparameter is the number of filters, which controls the number of distinct features that can be extracted and directly determines the depth of the output feature volume. The second hyperparameter is the spatial size of the filters (e.g., 3×3, 5×5), which specifies the receptive field over which local patterns are learned. The third hyperparameter is the stride, indicating the step size with which the filters slide across the input feature maps during convolution. Finally, zero-padding is applied around the input to regulate the spatial dimensions of the output feature maps and to alleviate boundary information loss caused by the convolution operation. If we have a volume of size $I_w \times I_h \times D_1$ then the output volume produced by the convolutional layer is of size $V_w \times V_h \times D_2$ and it is calculated using Eqs. 2, 3 and 4

$$V_w = \frac{I_w - F_w + 2P}{S} + 1 \tag{2}$$

$$V_h = \frac{I_h - F_h + 2P}{S} + 1 \tag{3}$$

$$D_2 = K \tag{4}$$

where F_w and F_h are the width and height of the filter. P, S and K represents the zero-padding, the stride and the number of filters, respectively.

(B) Pooling layer

The pooling operation, [29], is commonly introduced between successive convolutional layers in a convolutional neural network. It takes as input the feature maps generated by a convolutional layer

and aims to reduce their spatial resolution while retaining the most salient information. By decreasing the dimensionality of the feature maps, pooling layers significantly reduce the number of trainable parameters, improve computational efficiency, lower memory consumption, and help mitigate overfitting.

The output feature maps produced by a pooling layer can be expressed as follows in Eq. 5:

$$P_j^l = \vartheta\left(\beta_j^l \cdot \text{down}(C_i^{l-1}) + b_j^l\right) \tag{5}$$

where C_i^{l-1} denotes the i^{th} feature map in the $(l-1)^{th}$ layer, and down$(\cdot)$ represents the pooling (downsampling) function. The parameter β_j^l corresponds to the weighting coefficient associated with the j^{th} feature map in the l^{th} layer, while b_j^l denotes the bias term. The resulting pooled feature map at the l^{th} layer is denoted by P_j^l.

Several pooling strategies have been proposed in the literature, including max pooling, average pooling, and sum pooling. Max pooling retains the maximum value within each local spatial region, average pooling computes the mean value over the pooling window, and sum pooling aggregates all values within the region. Among these approaches, max pooling is the most widely adopted due to its effectiveness in capturing dominant features. The pooling layer is primarily controlled by two hyperparameters: the filter size (F), which defines the spatial extent of the pooling window, and the stride (S), which determines the step size used to slide the pooling window across the feature map. If we have a volume of size $I_w \times I_h \times D_1$, then the output volume produced by a pooling layer is of size $V_w \times V_h \times D_2$, computed as shown in Eqs. 6, 7 and 8:

$$V_w = \frac{I_w - F}{S} + 1 \tag{6}$$

$$V_h = \frac{I_h - F}{S} + 1 \tag{7}$$

$$D_2 = D_1 \tag{8}$$

(C) Activation function

To introduce nonlinearity into the network, each output from the convolutional layer is passed through a nonlinear activation function. In this study, the Rectified Linear Unit (ReLU), [30], is chosen due to its straightforward implementation and high efficiency. ReLU transforms all negative input values to zero while leaving positive values unchanged, allowing the model to capture complex patterns more effectively.

A key benefit of using ReLU is that it can speed up the training process and help mitigate the vanishing gradient issue, which often hinders deep network learning. Mathematically, the ReLU function can be expressed as in Eq. 9:

$$ReLU(x) = \max(0, x) \tag{9}$$

(D) Batch Normalization

Batch normalization (BN), [31], plays an important role in enhancing the stability and efficiency of deep neural networks. The method works by standardizing the inputs of each layer so that they have a mean of zero and a variance of one. This normalization reduces internal covariate shifts, stabilizes the training process, and often accelerates convergence. Additionally, batch normalization can provide a regularizing effect, which helps in minimizing overfitting during training.

Let $x \in \mathbb{R}^{H\times W\times N\times C}$ be an activation tensor in a convolutional neural network, where the axes H, W, N and C represent the height and the width, the batch and the number of channels, respectively. We set B, the subset resulting from the flattening of x on all axes except that of the channels.

For any activation x_i in $B = \{x_1, x_2, ..., x_m, m \in [1, N] \times [1, H] \times [1, W]\}$, the standardization with BN is written as in Eq. 10:

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{10}$$

where the mean μ_B and the variance σ_B^2 of B are given by Eqs. 11 and 12, while $\epsilon > 0$ is a small value which handles numerical instabilities.

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{11}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \tag{12}$$

Finally, two parameters γ and β are introduced to increase network nonlinearity; the output y of the BN layer is given by Eq. (13)

$$y_i = \gamma\hat{x}_i + \beta \tag{13}$$

where γ is the scale parameter and β is the shift parameter (E) Dense Layer

In our approach, the multi-dimensional feature maps produced by the previous layers are first converted into a single long vector using a flattening operation. This transformation allows the extracted information to be processed by a fully connected network. The resulting vector is denoted as $p =$

$\{p_1, p_2, \ldots, p_I\}$, where I is the total number of elements from the final pooling layer.

This vector is then fed into a shallow network consisting of three fully connected layers to perform the final classification. The output of each neuron in these layers is given by Eq. 14:

$$y_j^l = \vartheta\left(\sum_{i \in l-1} w_{i,j}^l \, p_i^{l-1} + b_j^l\right) \tag{14}$$

where y_j^l is the output of the j-th neuron in layer l, and $\vartheta(\cdot)$ represents the chosen activation function.

In our design, ReLU is applied to the hidden layers to introduce nonlinearity, while the last layer uses the softmax function to generate a probability distribution across all classes. Each neuron in a fully connected layer integrates information from every neuron in the preceding layer, allowing the network to combine and interpret complex patterns from the input. The softmax output is defined by Eq. 15:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \quad i = 1, \ldots, K \tag{15}$$

where $z = [z_1, z_2, \ldots, z_K]$ represents the inputs to the final layer, K is the number of target classes, and the resulting probabilities satisfy $\sum_{i=1}^{K} softmax(z_i) = 1$. For this study, the model predicts three classes.

(F) Dropout layer

Dropout, [32], is a powerful technique used to regularize the training of deep network. This technique is intended to prevent overfitting to training data. Indeed, deep neural networks are composed of several non-linear hidden layers.Including deeper and more expressive layers increases the model's capacity to approximate complex input–output functions. Nevertheless, this flexibility may cause the network to adapt to incidental patterns present only in the training data, which do not necessarily persist when the model is evaluated on unseen data. When this happens, the model does not generalize well, resulting in poor classification performance. This consists of forcing the neural network to learn several independent representations of the same data (abandoning neurons during the learning phase), with the aim of reducing dependence on the training set and avoiding co-adaptations complex data on the training sample. The term "dropout" refers to a temporary and random deactivation of neurons and all its incoming and outgoing connections in the network with probability p. The latter is thus deprived of part of its neurons during the training phase (their value is considered to be 0) then they will be reactivated once again to test the model.

Formally, training with Dropout for layer l is expressed as Eqs. 16, 17 and 18:

$$r_j^l \sim \text{Bernoulli}(p) \tag{16}$$

$$\tilde{x}^l = r_j^l \odot x^l \tag{17}$$

$$x_i^{l+1} = \vartheta\left(w_i^{l+1} \odot \tilde{x}^l + b_i^{l+1}\right) \tag{18}$$

where x^l denotes the original output of layer l, and $\tilde{x}^l$ represents the modified output after applying dropout. The symbol $\circledast$ indicates element-wise multiplication. For any given layer l, r^l is a vector of independent Bernoulli random variables, each taking the value 1 with probability p.

(G) Loss Function

The aim of network training is to minimize the error between the network prediction and the actual value. The loss function used in our model is the categorical cross-entropy loss. This is stated mathematically as Eq. 19:

$$\mathcal{L}(p, q) = -\sum_{k} p_k \log q_k \tag{19}$$

where, k is the number of classes, q_k is the predicted probability of class k or Softmax function and p_k denotes the true output.

(H) The proposed MRC Block

In traditional neural networks, having more layers means a better network, but due to the vanishing gradient problem, the weights of the first layers of the network will not be updated correctly by backpropagation. As the error gradient is backpropagated to previous layers from the output layer, repeated adjustments of the weights by these decreasing differential factors make the gradient smaller and smaller. So, with more layers in the networks, the performance of the training process becomes saturated and decreases quickly. The Res-Net network, solves this problem by adding residual blocks which allow the inputs of a layer to be connected directly to the output of this same layer by a skip connection. Links transmitted directly from the input to the output of such residual blocks will be assigned unit weights. In this type of architecture, when backpropagation is carried out, the gradient will be weighted by the weights of the so-called residual connections which are 1. This approach then preserves the entries and avoids any loss of information.

Figure 1(a) (Appendix) depicts a residual block, which includes two convolutional layers with 3×3 kernels. Each layer is followed by batch normalization and a ReLU activation. The output of

the second convolution is merged with the block's original input, and the combined result is passed through an additional ReLU function to produce the final output. However, despite their success, standard residual blocks often struggle to efficiently capture multi-resolution features, especially when dealing with tasks that require simultaneous learning of fine details and global patterns. To overcome these limitations, we propose the MRC block, which enhances the traditional structure by introducing additional operations tailored for improved feature extraction and representation.

The MRC block builds on the residual block concept by incorporating a dual-path design, featuring a convolutional path for feature refinement and a skip connection path for efficient downsampling and feature alignment. One of the primary innovations in the MRC block is the integration of 1×1 convolutional layer and a 2×2 max pooling operation into the skip connection path. These modifications address the limitations of standard residual blocks by enabling the network to downsample feature maps while preserving critical information and expanding the number of feature channels, enhancing the learning capacity of the network.

As shown in Figure 1(b) (Appendix), the block is composed of two parallel paths. The left path is designed to extract detailed and abstract features by applying successive convolutional operations. This path begins with a 5×5 convolutional layer with a stride of 2, which serves to downsample the input feature map by reducing its spatial dimensions while increasing the number of channels. This operation is critical for capturing more abstract representations of the input data. The output of the first convolution is normalized using batch normalization, activated using the ReLu function (Eq. (20)), and regularized with dropout (Eq. (21)) to prevent overfitting and improve generalization. The first convolution is followed by a 3x3 convolutional layer with a stride of 1, which strengthens the extracted features without altering the spatial dimensions of the feature map. This layer ensures that the refined features are better suited for downstream tasks. The second convolutional layer is followed by batch normalization to stabilize the training process and ensure consistency across feature maps. The output of the left path in the MRC block, denoted as x_3^i, can be mathematically expressed as 22:

$$x_1^i = BN(\varphi(w^k \otimes_s x^{i-1} + b^k)), \quad \forall\, 1 \leq k \leq K,\ w^k \in \theta_1^i,\ b^k \in \theta_1^i \quad (20)$$

$$x_2^i = \text{Dropout}(x_1^i) \quad (21)$$

$$x_3^i = BN(\varphi(w^k \otimes_s x_2^i + b^k)), \quad \forall\, 1 \leq k \leq K,\ w^k \in \theta_3^i,\ b^k \in \theta_3^i \quad (22)$$

where x^{i-1} is the input feature map from the previous layer $i-1$, K is the number of kernels, θ is the set of parameters, including weight and biases for all kernels in the layer, w^k is the k^{th} convolution kernel of filter of size $m \times m$, applied to the input feature map, $\otimes$ is the convolution operation which shifts the kernel over the input feature map with stride s, b^k is the bias term associated with the k^{th} kernel, $BN(.)$ is the batch normalization operation and $\varphi(.)$ is the ReLu activation function.

The right path (skip connection path) complements the convolutional path by preserving essential features from the input while performing efficient downsampling. The first operation in this path is a 1x1 convolutional layer with a stride of 1, which adjusts the number of channels in the input feature map to match those of the convolutional path. This alignment ensures that the outputs of both paths are compatible for the merging operation. The output of the 1x1 convolution is normalized using batch normalization for stability (Eq. (23)). Following the 1x1 convolution, the feature map undergoes a 2x2 max pooling operation with a stride of 2. This operation reduces the spatial dimensions of the feature map by extracting the maximum value from each 2x2 region. Max pooling not only performs downsampling but also retains the most critical features, discarding redundant or irrelevant information. The use of max pooling in the skip connection path ensures that the path remains computationally efficient while preserving key patterns in the input data. The output of the left path in the MRC block, denoted as x_5^i, can be mathematically expressed as 24:

$$x_4^i = BN(\varphi(w^k \otimes_s x^{i-1} + b^k)), \quad \forall\, 1 \leq k \leq K,\ w^k \in \theta_4^i,\ b^k \in \theta_4^i \quad (23)$$

$$x_5^i = \text{pool}(x_4^i) \quad (24)$$

where $pool(.)$ is the pooling layer

After processing the input through the convolutional and skip connection paths, the outputs of both paths are merged using an element-wise addition operation. Mathematically, this is defined as Eq. 25:

$$x_6^i = x_5^i \oplus x_3^i \quad (25)$$

where $\oplus$ represents the element-wise addition operation.

This merging step integrates the detailed and refined features from the convolutional path with

the compact and essential features from the skip connection path. The merged output is then passed through a ReLU activation function, which introduces non-linearity and improves the network's ability to learn complex relationships in the data.The output of the MRC block is given by Eq. 26:

$$x^i = \varphi(x_6^i) \tag{26}$$

This design provides significant advantages, including improved accuracy due to better feature representation, efficient gradient propagation that enables the training of deeper networks, and computational efficiency through lightweight operations in the skip connection path. Additionally, the MRC block offers versatility, allowing seamless integration into various deep learning architectures for tasks like classification. Its ability to downsample inputs, expand channels, and extract multi-scale features makes it a robust and effective building block for modern neural networks.

The implementation details of the proposed COVID-MRSNet model

The network receives as input chest x-ray images of size 256×256×1 and generates a vector for each image with one input for each class. As depicted in Figure 2 (Appendix) and listed in Table 1 (Appendix), the introduced network consists of input images, feature extraction section and classification layer. This network contains three major parts. The first part is a combination of two-dimensional convolutional layer (Conv2D), batch normalization (BN) and ReLU activation function. The BN and ReLu activation function was applied on the conv2D outputs. We used the ReLU function to improve nonlinearity and batch normalization to increase the stability of the model.

At the 1st layer of Conv2D, we used 64 filters to learn from input with a 7×7 convolution kernel size; 1 stride and with padding = 'same'. Padding = 'same' in conv2D means padding with zeros uniformly so that the output has the same dimension as the input and so that information is not lost at the edges. The second part of this network is the repetitive of the proposed MRC block which consists of 5×5 Conv2D with a stride of 2 and without padding, BN, ReLu activation function, Dropout, 3×3 Conv2D with a stride of 1 and padding = 'same', BN, respectively. Another parallel path exists next to this step, called skip connection, which helps the block learn better. In this skipped connection of the residual block, we used 1×1 convolutional layer with a stride of 1 and padding='same' followed by a batch normalization layer and 2×2 Max pooling layer with a stride of 2. Then, the output of this parallel step will be added to the previous result. After this, a ReLu activation function is applied. Before adding the two parallel paths, we use zero padding technique to keep the output size dimension of the first path same as the output size dimension of the second one. In these five blocks, the number of filters increases from 64 to 1024 and the dropout value changes from 0.1 to 0.5. At the 1st block, all layers of Conv2D use 64 filters and a dropout value of 0.1, at the 2nd block, all layers of Conv2D use 128 and a dropout value of 0.2, at the 3rd block, all layers of Conv2D use 256 filters and a dropout value of 0.3, at the 4th block, all layers of Conv2D use 512 filters and a dropout value of 0.4; and at the 5th block, all layers of Conv2D use 1024 filters and a dropout value of 0.5. The reason behind applying different numbers of filters was to extract multiple and dominant features and behind applying different dropout values was to reduce overfitting. Moreover, the reason for applying different kernel size was to extract local features and more global information. The output outcomes of the MRC block (5) are followed by MaxPooling layer with 2×2 pooling size and 2 strides. The third part of this architecture is the classification layers which produce the final decision regarding the label of the input data. The outputs of the MaxPooling layer are flattened and entered into three dense layers. The first dense layer contains 512 neurons. This layer transmits the result to the second dense layer which has 256 neurons. These two first dense layers are followed by the ReLU activation function with L2 regularization. The final output of the proposed model is developed from a dense layer with three neurons and a Softmax function with L2 regularization, which classifies the output of Chest X-Ray image into one of the classes: COVID-19, viral pneumonia or normal. We set the value of the regularization parameter to 0.02. Regularization is necessary to avoid overfitting the model, and this is done by adding a penalty to the cost function.

The total number of parameters of the proposed COVID-MRSNet is 39,335,619, which consist of 39,323,715 trainable parameters and 11,904 non-trainable parameters

4 Experimental setup

The experiments were carried out using a core i7, GTX 1060 6GB GPU and programmed in the python language. Area under the receiver operating characteristic curve along with accuracy are used as the major performance metrics. The confusion matrix has been extracted. Precision, recall and F1-score are used as the auxiliary performance measures in predicting COVID-19 class. Moreover, the proposed COVID-MRSNet model is evaluated on 5-fold cross-validation. This section outlines a comprehensive view of the dataset and the classification performance metrics used in the study.

4.1 Performance evaluation metrics

To examine the performance of the COVID-MRSNet model, we applied a 5-fold cross-validation procedure. The dataset was randomly split into five equally sized segments, which were rotated to serve as training, validation, and test sets. In practice, 70% of the data was used for training, 10% for validation, and 20% for testing. The evaluation relied on commonly employed metrics, [33], including overall accuracy, the proportion of correctly predicted positive samples (precision), the model's ability to identify actual positives (recall), the combined F1-score, and the Area Under the Receiver Operating Characteristic curve (AUC). All these metrics were calculated using values from the confusion matrix, where TP and TN indicate correctly predicted positive and negative instances, while FP and FN correspond to incorrect predictions.

(A) Confusion Matrix

The confusion matrix, [34], summarizes how well the model predicts each class. Each row represents the true class, and each column corresponds to the predicted class. This matrix provides a comprehensive overview of classification outcomes and forms the basis for computing all subsequent performance metrics. The matrix size depends on the number of categories being classified.

(B) Precision

Precision reflects the fraction of instances classified as positive that were actually positive. It is defined as in Eq. 27:

$$Precision = \frac{TP}{TP + FP}. \tag{27}$$

High precision indicates that the model rarely mislabels negative cases as positive.

(C) Accuracy

Accuracy measures the proportion of correctly identified samples, considering both positive and negative classes. It is defined in Eq. 28:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{28}$$

(D) Recall

Recall (also called sensitivity) quantifies the fraction of actual positive cases that are correctly detected. It is calculated in Eq. 29:

$$Recall = \frac{TP}{TP + FN}. \tag{29}$$

Higher recall means the model misses fewer positive instances.

(E) F1-Score

The F1-score balances precision and recall by calculating their harmonic mean. It is given by Eq. 30:

$$F1\text{-}score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}. \tag{30}$$

It provides a single measure of performance when the dataset is imbalanced.

(F) ROC Curve and AUC

The ROC curve depicts the relationship between the true positive rate and the false positive rate across different decision thresholds. The AUC quantifies the model's overall ability to distinguish between classes, with values close to 1 representing strong discrimination and values near 0.5 indicating random guessing. For multi-class tasks, a separate ROC curve can be computed for each class, treating it as the positive case and all others as negative.

4.2 Dataset description and Preprocessing

The proposed COVID-MRSNet framework is trained, validated, and evaluated using the COVID-19 Radiography Database available on Kaggle, [35]. This publicly accessible dataset was compiled by researchers from Qatar University and the University of Dhaka, Bangladesh, in collaboration with institutions from Malaysia and Pakistan, as well as medical professionals.

The dataset is composed of three distinct classes of chest X-ray images: (1) 219 images corresponding to patients diagnosed with COVID-19, (2) 1,345 images of patients affected by viral pneumonia, and (3) 1,341 images from healthy individuals. All images are provided in PNG format with an original spatial resolution of 1024×1024 pixels.

For experimental evaluation, the dataset is partitioned into three mutually exclusive subsets, namely training, validation, and testing, containing 2,092 (70%), 232 (10%), and 581 (20%) images, respectively. During the pre-processing stage, each image is first cropped to remove non-clinical elements such as textual annotations, symbols, and artifacts that may interfere with feature learning. Subsequently, the cropped images are resized to a standardized resolution of 256×256 pixels using the Python Pillow library.

In addition, image normalization is applied to adjust pixel intensity distributions, thereby enhancing image consistency and improving training stability. These pre-processing steps contribute to reducing computational complexity, accelerating the training process, and lowering random access memory (RAM) consumption. The distribution of images across the three classes (COVID-19, viral pneumonia, and normal) for the training, validation, and testing sets is summarized in Table 2 (Appendix). Representative

examples of chest X-ray images from each category are illustrated in Figure 3 (Appendix).

5 Experimental Results and discussion

This section provides and discusses the experimental results obtained by our proposed COVID-MRSNet presented in Section 3 for COVID-19 classification.

5.1 Results

The proposed COVID-MRSNet model is designed to categorize CXR images into three classes: COVID-19, pneumonia, and normal (no findings). Its performance was evaluated using two commonly applied approaches. The first approach involves dividing the dataset into five independent subsets. Four of these subsets are utilized for training and validation, while the remaining subset is reserved for testing. The second approach employs a five-fold cross-validation strategy to evaluate the model's capability in the three-class classification task. For each fold, performance was measured using the confusion matrix and the metrics outlined in Section 4.

Across all experiments, the model was trained for 50 epochs using the Adam optimizer. The learning rate was set to 0.00001 and the batch size to 32. These hyperparameters, including the number of epochs, batch size, and learning rate, were selected based on empirical experimentation. A complete summary of all model parameters is presented in Table 3 (Appendix).

Figure 4 (Appendix) presents the confusion matrices for each fold in the three-class classification task. Overall, the model demonstrates high true positive rates across all classes.

As illustrated in Figure 4(a) (Appendix), out of 43 COVID-19 chest X-ray images, COVID-MRSNet correctly identified 42 cases and misclassified one case as viral pneumonia (non-COVID-19). For pneumonia images, the model accurately classified 267 out of 269 cases, with 2 instances misclassified as normal. Regarding healthy chest X-rays, 260 out of 269 normal cases were correctly identified, while 1 case was misclassified as COVID-19 and 8 cases as viral pneumonia.

In Figure 4(b) (Appendix), all COVID-19 cases were correctly classified. Among normal cases, 8 were misclassified as pneumonia and 1 as COVID-19. Only 2 pneumonia cases were incorrectly predicted as normal.

Figure 4(c) (Appendix) shows that the network predicted all COVID-19 cases correctly. For 269 normal images, 264 were accurately classified (classification rate of 98.14%), with 5 misclassified as pneumonia. Pneumonia cases were correctly identified in 266 out of 269 instances (classification rate of 98.88%), with 3 cases misclassified as normal.

From Figure 4(d) (Appendix), it can be seen that all COVID-19 cases were correctly recognized. Misclassification occurred for 2.23% of normal cases as pneumonia and 1.49% of pneumonia cases as normal. These results indicate that COVID-MRSNet consistently distinguishes COVID-19 cases from non-infected cases across nearly all folds.

Finally, Figure 4(e) (Appendix) demonstrates that the model predicted all COVID-19 cases correctly. Normal images were accurately classified in 263 cases (97.77%), with 1 misclassified as COVID-19 and 5 as viral pneumonia. For pneumonia, 264 out of 269 cases were correctly identified, with 5 misclassified as normal.

The precision, recall, and F1-score for each class (Normal, COVID-19, and Viral Pneumonia) across all folds are summarized in Table 4 (Appendix), Table 5 (Appendix), Table 6 (Appendix), Table 7 (Appendix), and Table 8 (Appendix). Overall, the results demonstrate the strong performance of the proposed model for all categories.

Table 4 (Appendix) shows the results for fold-1. For COVID-19 cases, the model achieved a precision of 97.67%, while the recall reached 97.67% and the F1-score was also 97.67%. Normal cases were classified with 99.24% precision. The recall for normal cases was 96.65% and the F1-score reached 97.93%. Viral pneumonia cases showed 96.74% precision, 99.26% recall, and 97.98% F1-score.

In fold-2 (Table 5) (Appendix), the COVID-MRSNet model achieved perfect performance for COVID-19 across all three metrics. For normal cases, precision was measured at 98.88%, recall was 98.51%, and F1-score reached 98.69%. Viral pneumonia cases recorded a precision of 98.52%, recall of 98.88%, and an F1-score of 98.70%.

Fold-3 results are summarized in Table 6 (Appendix). COVID-19 cases again reached 100% for precision, recall, and F1-score. Normal cases achieved 98.87% precision, 98.14% recall, and 98.51% F1-score. For viral pneumonia, the network produced 98.16% precision, 98.89% recall, and 98.52% F1-score.

Table 7 (Appendix) presents the outcomes for fold-4. COVID-19 detection maintained perfect results across all three metrics. Normal cases showed a precision of 98.50%, recall of 97.77%, and F1-score of 98.13%. Viral pneumonia cases achieved 97.78% precision, 98.51% recall, and 98.15% F1-score.

Finally, fold-5 results are reported in Table 8 (Appendix). COVID-19 cases obtained a precision of 97.72%, a recall of 100%, and an F1-score of 98.85%.

Viral pneumonia cases recorded 98.14% for precision, 98.14% for recall, and 98.14% F1-score. Normal cases reached 98.13% precision, 97.77% recall, and 97.95% F1-score.

In summary, the proposed CNN consistently delivers high performance across all folds. These results confirm that the model can reliably distinguish between COVID-19, viral pneumonia, and normal cases, demonstrating its robustness and effectiveness in multi-class classification.

Table 9 (Appendix) summarizes the numerical results of cross-validation of the introduced framework for multi-class categorization. At first, we demonstrated the performance for each fold, and then we calculated the mean of the five folds. As can be seen from the table, COVID-MRSNet model performed good results and reached 5-fold cross validation mean accuracy, precision, f1-score and recall of 98.34%, 98.55%, 98.61%, and 98.67%, respectively. It can be seen that the proposed COVID-MRSNet model has high accuracy, precision, f1-score and recall and has good generalization performance.

Figure 5 (Appendix), Figure 6 (Appendix), Figure 7 (Appendix), Figure 8 (Appendix), and Figure 9 (Appendix) display the performance of suggested COVID-MRSNet classifier in terms of loss and accuracy during the training and validation phases in each fold trained during 50 epochs.

From Figure 5 (Appendix), we observe that the highest obtained training and validation accuracies at fold-1 of the proposed CNN model was 1 and 0.9871, respectively. Moreover, the training loss was 0.0630 and validation loss was 0.1029 for 3-class classification.

From Figure 6 (Appendix), we see that the highest obtained training and validation accuracies at fold-2 of our model was 1 and 0.9914, respectively. The training loss was 0.0543 and validation loss was 0.0879 for 3-class classification.

From Figure 7 (Appendix), we notice that the training and validation accuracies at fold-3 of the COVID-MRSNet model is increased from 0.75 to 1 and from 0.8707 to about 0.99, respectively. Furthermore, the training loss is decreased from 1.8850 to about 0.05 and validation loss is decreased from 1.5429 to about 0.07.

From Figure 8 (Appendix), we observe that the training and validation accuracies at fold-4 of the COVID-MRSNet model is increased from 0.7557 to 1 and from 0.7802 to about 0.98, respectively. Moreover, the training loss is decreased from 1.9809 to about 0.05 and validation loss decreased from 1.8558 to about 0.09.

As illustrated in Figure 9 (Appendix), at the 50th epoch, the model achieves a training accuracy of 1.0 and a validation accuracy of 0.9871. Correspondingly, the training and validation losses are 0.0881 and 0.1209, respectively. These results indicate that increasing the number of training epochs leads to a steady decrease in training loss and a consistent improvement in training accuracy.

Overall, the proposed model demonstrates effective learning and can accurately classify chest X-ray images into COVID-19, pneumonia, and normal categories. The training curves also suggest that the model avoids both overfitting and underfitting, which accounts for the strong performance observed in precision, recall, F1-score, accuracy, and the confusion matrix.

Another metric used in verifying the performance of our model for multi-class classification is the ROC curve. An AUC value close to one indicates that the classification performance of the model is high. Figure 10 (Appendix) illustrates the Receiver Operating Characteristic (ROC) curve and presents the macro, micro, and class-wise AUC values obtained in each fold using the proposed COVID-MRSNet for multi-class classification. It can be observed that the AUC scores across all classes are relatively consistent, with the AUC for COVID being slightly higher than the other. In ROC curves, we obtained at fold-1 AUC scores of 1, 0.9990 and 0.9991 for COVID-19, viral pneumonia and normal, respectively. At fold-2, we reached AUC scores of 1, 0.9994 and 0.9994 for COVID-19, viral pneumonia and normal, respectively. At fold-3, we achieved AUC scores of 1, 0.9999 and 0.9999 for COVID-19, viral pneumonia and normal, respectively. At fold-4, we obtained AUC scores of 1, 0.9993 and 0.9993 for COVID-19, viral pneumonia and normal, respectively. At fold-5, AUC scores are 1, 0.9996 and 0.9996 for COVID-19, viral pneumonia and normal, respectively.

Overall, the proposed CNN network showed a remarkable AUC score of 1.0 on COVID-19 images.

5.2 Analysis and Benchmarking Against Existing Methods

In this study, a deep CNN is presented for the automatic identification of COVID-19 from chest X-ray images. The proposed framework addresses a three-class classification problem by differentiating COVID-19 cases from viral pneumonia and normal chest X-ray samples. A notable contribution of this work is that the proposed COVID-MRSNet architecture processes CXR images directly, without requiring handcrafted or manually engineered feature extraction steps. Model evaluation relies on commonly used performance indicators.

Table 10 (Appendix) provides a comparative analysis between the proposed COVID-MRSNet

classifier and several state-of-the-art deep CNN-based approaches reported for three-class COVID-19 classification. The comparative results indicate that COVID-MRSNet consistently demonstrates stronger performance than many existing methods summarized in the table. Nonetheless, several competitive solutions have also been described in recent studies. For instance, the work reported in [36], analyzed different CNN architectures for COVID-19 detection using chest X-ray images and identified XCeption as the most effective model. The reported outcomes include a precision reported as 97%, an F1-score reaching 86%, and a recall measured at 78.0%, while the overall classification accuracy reached 97.97%.

In another investigation, [15], the proposed deep learning-based approach achieved a recall measured at 96.9%, accompanied by an F1-score reaching 95.6%. In the same study, the precision was reported as 95%, whereas the classification accuracy reached 95%. Similarly, the model presented in [37], demonstrated strong classification capability, reporting recall and F1-score values both reaching 96.26%, with precision measured at 96.28%, while the corresponding accuracy reached 96.07%.

The authors of [38], introduced CXGNet, a deep CNN architecture designed for COVID-19 classification from chest X-ray images. Their experimental results indicate a recall reaching 96.96%, an F1-score measured at 95.38%, and a precision reported as 94.44%, while the achieved classification accuracy reached 97.05%.

Furthermore, [8], proposed COVID-Net, an automated deep learning framework in which the average classification accuracy reached 93.3% for the three-class task. In contrast, the capsule network-based approach described in [16], demonstrated a lower average accuracy reaching 84.22%.

The EfficientNet-based framework introduced in [39], achieved a recall measured at 97.54% and an F1-score reaching 97.11%. The reported precision reached 96.69%, while the resulting classification accuracy reached 96.7% for the discrimination between COVID-19, pneumonia, and normal cases.

The CoroDet model presented in [24], reported a recall measured at 92.5% and an F1-score reaching 91.32%, accompanied by a precision reported as 94.04%. The corresponding classification accuracy reached 94.2% for the three-class scenario.

For the DarkNet-based architecture introduced in [7], the reported performance includes a recall measured at 91.39% and an F1-score reaching 90.29%, followed by a precision reported as 88.84%, while the achieved average accuracy reached 86.37%.

Additionally, the approach proposed in [40], demonstrated a recall measured at 92.39% and an F1-score reaching 93.38%. In the same study, precision was reported as 89.46%, while the average accuracy reached 90.39%.

Overall, the accuracy values associated with the competing methods summarized in Table 10 (Appendix) range from 84.22% to 97.97%. When compared against these existing approaches, the proposed COVID-MRSNet framework achieves the strongest overall performance across all evaluation metrics, including accuracy, precision, recall, and F1-score. These results highlight the robustness and reliability of COVID-MRSNet as a decision-support system for assisting medical professionals. Although the proposed framework is not intended to replace clinical judgment, it can provide valuable support for large-scale screening and help reduce the exposure risk faced by healthcare workers during pandemic situations.

6 Conclusion and Future direction

Timely identification of COVID-19 cases is essential for controlling disease transmission. In this paper, a novel, powerful and fully automatic deep CNN called COVID-MRSNet is introduced for COVID-19 disease detection using chest X-Ray images. The developed model was trained on an openly available collection of CXR images, organized into three distinct classes. Various performance evaluation metrics were used to evaluate the effectiveness of the proposed model. The COVID-MRSNet model attains an average accuracy of 98.34%, alongside precision, recall, and F1-score values of 98.55%, 98.67%, and 98.61%, respectively, using k-fold cross-validation for three classes (COVID vs. Pneumonia vs. Normal. Performance classification show that the proposed COVID-MRSNet model achieved promising results and yielded the highest accuracy compared to the existing deep CNN mentioned in this study. Overall, the proposed network significantly advances in image-based radiological approaches and shows strong potential as a supportive tool for healthcare professionals, enabling more effective identification, assessment, and follow-up of COVID-19 patients. In subsequent work, we plan to adapt the network for a broader ranger of pulmonary conditions.

References:

[1] Tao, A. and Zhenlu, Y. and Hongyan, H. and Chenao, Z. and Chong, C. and Wenzhi, L. and Qian, T. and Ziyong, S. and Liming, X. (2020) Correlation of chest CT and RT-PCR testing in coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases, *Radiology*, Vol. 296, No. 2, p. 200642. https://doi.org/10.1148/radiol.2020200642

[2] Kanne, J. P., Little, B. P., Chung, J. H., Elicker, B. M., and Ketai, L. H. (2020) Essentials for radiologists on COVID-19: an update-radiology scientific expert panel, *Radiology*, Vol. 296, No. 2, pp. E113–E114. https://doi.org/10.1148/radiol.2020200527

[3] Bai, H. X., Hsieh, B., Xiong, Z., Halsey, K., Choi, J. W., Tran, T. M. L., Pan, I., Shi, L.-B., Wang, D.-C., Mei, J., Jiang, X.-L., Zeng, Q.-H., Egglin, T. K., Hu, P.-F., Agarwal, S., Xie, F.-F., Li, S., Healey, T., Atalay, M. K., Liao, W.-H. (2020) Performance of radiologists in differentiating COVID-19 from non-COVID-19 viral pneumonia at chest CT, *Radiology*, Vol. 296, No. 2, pp. E46–E54. https://doi.org/10.1148/radiol.2020200823

[4] Fang, Y., Zhang, H., Xie, J., Lin, M., Ying, L., Pang, P., and Ji, W. (2020) Sensitivity of chest CT for COVID-19: comparison to RT-PCR, *Radiology*, Vol. 296, No. 2, p. 200432. https://doi.org/10.1148/radiol.2020200432.

[5] Bernheim, A., Mei, X., Huang, M., Yang, Y., Fayad, Z. A., Zhang, N., Diao, K., Lin, B., Zhu, X., Li, K., Li, S., Shan, H., Jacobi, A., Chung, M. (2020) Chest CT findings in coronavirus disease-19 (COVID-19): relationship to duration of infection, *Radiology*, Vol. 295, No. 3, pp. 685–691, https://doi.org/10.1148/radiol.2020200463.

[6] Long, C., Xu, H., Shen, Q., Zhang, X., Fan, B., Wang, C., Zeng, B., Li, Z., Li, X., Li, H., Li, H. (2020) Diagnosis of the coronavirus disease (COVID-19): RT-PCR or CT?, *Eur J Radiol*, Vol. 126, p. 108961. https://doi.org/10.1016/j.ejrad.2020.108961.

[7] Ozturk, T., Talo, M., Yildirim, E. A., Baloglu, U. B., Yildirim, O., Rajendra, A. (2020) Automated detection of COVID-19 cases using deep neural networks with X-ray images, *Comput Biol Med*, Vol. 121, p. 103792. https://doi.org/10.1016/j.compbiomed.2020.103792

[8] Wang, L., Lin, Z. Q., Wong, A. (2020) COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images, *Scientific Reports*, Vol. 10, p. 19549. https://doi.org/10.1038/s41598-020-76550-z.

[9] Demir, F. (2021) DeepCoroNet: a deep LSTM approach for automated detection of COVID-19 cases from chest X-ray images, *Soft Computing*, Vol. 103, p. 107160. https://doi.org/10.1016/j.asoc.2021.107160

[10] Hasan, K., Alam, A., Elahi, T. E., Roy, S., Wahid, S. R. (2020) CVR-Net: a deep convolutional neural network for coronavirus recognition from chest radiography images, Preprint at https://doi.org/10.48550/arXiv.2007.11993.

[11] Singh, D., Kumar, V., and Kaur, M. (2020) Densely connected convolutional networks-based COVID-19 screening model, *Applied Intelligence*, Vol. 51, pp. 3044–3051. https://doi.org/10.1007/s10489-020-02149-6

[12] Simonyan, K., and Zisserman, A. (2015) Very deep convolutional networks for large-scale image recognition, In: Proceedings of the 3rd International Conference on Learning Representations ICLR, San Diego, CA, USA, pp. 1–14. https://doi.org/10.48550/arXiv.1409.1556

[13] He, K., Zhang, X., Ren, S., and Sun, J. (2016) Deep residual learning for image recognition, In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp. 770–778. https://doi.org/10.1109/CVPR.2016.90

[14] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017) Densely connected convolutional networks, In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp. 4700–4708. https://doi.org/10.48550/arxiv.1608.06993

[15] Khan, A. I., Shah, J. L., and Bhat, M. M. (2020) CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images, *Comput Methods Programs Biomed*, Vol. 196, p. 105581. https://doi.org/10.1016/j.cmpb.2020.105581

[16] Toraman, S., Alakus, T. B., and Turkoglu, I. (2020) Convolutional CapsNet: a novel artificial neural network approach to detect COVID-19 disease from X-ray images using capsule networks, *Chaos, Solitons Fractals*, Vol. 140, p. 110122. https://doi.org/10.1016/j.chaos.2020.110122

[17] Abraham, B., and Nair, M. (2020) Computer-aided detection of COVID-19 from X-ray images using multi-CNN and Bayesnet classifier, *Biocybernetics and Biomedical*

Engineering, Vol. 40, pp. 1436–1445. https://doi.org/10.1016/j.bbe.2020.08.005

[18] Oh, Y., Park, S., and Ye, J. C. (2020) Deep Learning COVID-19 Features on CXR Using Limited Training Data Sets. *IEEE Transactions on Medical Imaging*, Vol. 39, No. 8, pp. 2688–2700. https://doi.org/10.1109/TMI.2020.2993291

[19] Li, L., Qin, L., Xu, Z., Yin, Y., Wang, X., Kong, B., Bai, J., Lu, Y., Fang, Z., Song, Q., Cao, K., Liu, D., Wang, G., Xu, Q., Fang, X., Zhang, S., Xia, J., Xia, J. (2020) Using Artificial Intelligence to Detect COVID-19 and Community-acquired Pneumonia Based on Pulmonary CT: Evaluation of the Diagnostic Accuracy, *Radiology*, Vol. 296, pp. E65–E71. https://doi.org/10.1148/radiol.2020200905

[20] Afshar, P., Heidarian, S., Naderkhani, F., Oikonomou, A., Plataniotis, K. N., and Mohammadi, A. (2020) Covid-Caps: A Capsule Network-Based Framework for Identification of COVID-19 Cases From X-ray Images, *Pattern Recognition Letters*, Vol. 138, pp. 638-643. https://doi.org/10.1016/j.patrec.2020.09.010

[21] Hemdan, E. E.-D., Shouman, M. A., and Karar, M. E. (2020) Covidx-net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-ray Images, Preprint at https://doi.org/10.48550/arXiv.2003.11055.

[22] Ouchicha, C., Ammor, O., and Meknassi, M. (2020) Cvdnet: a novel deep learning architecture for detection of coronavirus (COVID-19) from chest X-ray images, *Chaos Solitons Fractals*, Vol. 140, p. 110245. https://doi.org/10.1016/j.chaos.2020.110245

[23] Shibly, K. H., Dey, S. K., Islam, M. T. U., and Rahman, M. M. (2020) Covid faster r-cnn: a novel framework to diagnose novel coronavirus disease (COVID-19) in X-ray images, *Informatics in Medicine Unlocked*, Vol. 20, p. 100405. https://doi.org/10.1016/j.imu.2020.100405

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

[25] Gilanie, G., Bajwa, U. I., Waraich, M. M., Asghar, M., Kousar, R., Kashif, A., Aslam, R. S., Qasim, M. M., Rafique, H. (2021) Coronavirus (COVID-19) detection from chest radiology images using convolutional neural networks, *Biomed Signal Process Control*, Vol. 66, p. 102490. https://doi.org/10.1016/j.bspc.2021.102490

[26] Farooq, M., and Hafeez, A. (2021) Covid-ResNet: A deep learning framework for screening of Covid-19 from radiographs, Preprint at https://doi.org/10.48550/arXiv.2003.14395.

[27] Oren, O., Gersh, BJ., Bhatt, DL. (2020) Artificial intelligence in medical imaging: switching from radiographic pathological data to clinically meaningful endpoints. *Lancet Digit Health*, Vol. 2, No. 9, p. e486–e488. https://doi.org/10.1016/S2589-7500(20)30160-6

[28] Asok, A. K., and Thampi, L. L. (2024) Automated Diagnosis of COVID-19 and Pneumonia Using Transfer Learning and Custom Segmentation on Chest X-Ray Images. *WSEAS Transactions on Computer Research*, Vol. 12, pp. 328–335. https://doi.org/10.37394/232018.2024.12.32

[29] Loussaief, S., Abdelkrim, A. (2018) A Convolutional neural network hyper-parameters optimization based on genetic algorithms, *International Journal of Advanced Computer Science and Applications*, Vol. 9, No. 10, pp. 252–266. http://doi.org/10.14569/IJACSA.2018.091031

[30] Nair, V., and Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. In Proceedings of the 27th International Conference on Machine Learning (ICML 2010), Haifa, Israel, pp. 807–814. https://dblp.org/rec/conf/icml/NairH10 (accessed July 20, 2010)

[31] Ioffe, S., and Szegedy, C. (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: Proceedings of the 32nd International Conference on Machine Learning (ICML 2015), Lille, France, pp. 448–456. https://proceedings.mlr.press/v37/ioffe15.html (accessed July 9, 2015)

[32] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov,

R. (2014) Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, Vol. 15, No. 1, pp. 1929–1958. http://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf (accessed April 30, 2014)

[33] Hossin, M., and Sulaiman, M. N. (2015) A review on evaluation metrics for data classification evaluations, *Int. J. Data Min. Knowl. Manag. Process*, Vol. 5, No. 2. https://doi.org/10.5121/ijdkp.2015.5201

[34] Narkhede, S. (2022) Performance Metrics: Confusion matrix, Precision, Recall, and F1 Score, *Medium*. https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62 (accessed May 23, 2022)

[35] Rahman, T., Chowdhury, M., Khandakar, A. (2020) COVID-19 CHEST RADIOGRAPHY DATABASE. Available online: https://www.kaggle.com/tawsifurrahman/covid19-radiography-database (accessed March 15, 2022)

[36] Jain, R., Gupta, M., Taneja, S., and Hemanth, D. J. (2021) Deep learning-based detection and analysis of COVID-19 on chest X-ray images, *Appl Intell.*, Vol. 51, No. 3, pp. 1690–1700. https://doi.org/10.1007/s10489-020-01902-1

[37] Chowdhury, N. K., Kabir, M. A., Rahman, M. M., and Rezoana, N. (2021) ECOVNet: a highly effective ensemble based deep learning model for detecting COVID-19, *PeerJ Comput Sci.*, Vol. 7, No. 5, p. e551. https://doi.org/10.7717/peerj-cs.551

[38] Gopatoti, A., and Vijayalakshmi, P. (2022) CXGNet: A tri-phase chest X-ray image classification for COVID-19 diagnosis using deep CNN with enhanced grey-wolf optimizer, *Biomed. Signal Process. Control*, Vol. 77, p. 103860. https://doi.org/10.1016/j.bspc.2022.103860

[39] Marques, G., Agarwal, D., and Torre, I. (2020) Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network, *Appl Soft Comput.*, Vol. 96, p. 106691. https://doi.org/10.1016/j.asoc.2020.106691

[40] Apostolopoulos, I. D., and Mpesiana, T. A. (2020) Covid-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks, *Phys. Eng. Sci. Med.*, Vol. 43, No. 2, pp. 635–640. https://doi.org/10.1007/s13246-020-00865-4

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The author contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The author has no conflict of interest to declare that is relevant to the content of this article.

APPENDIX

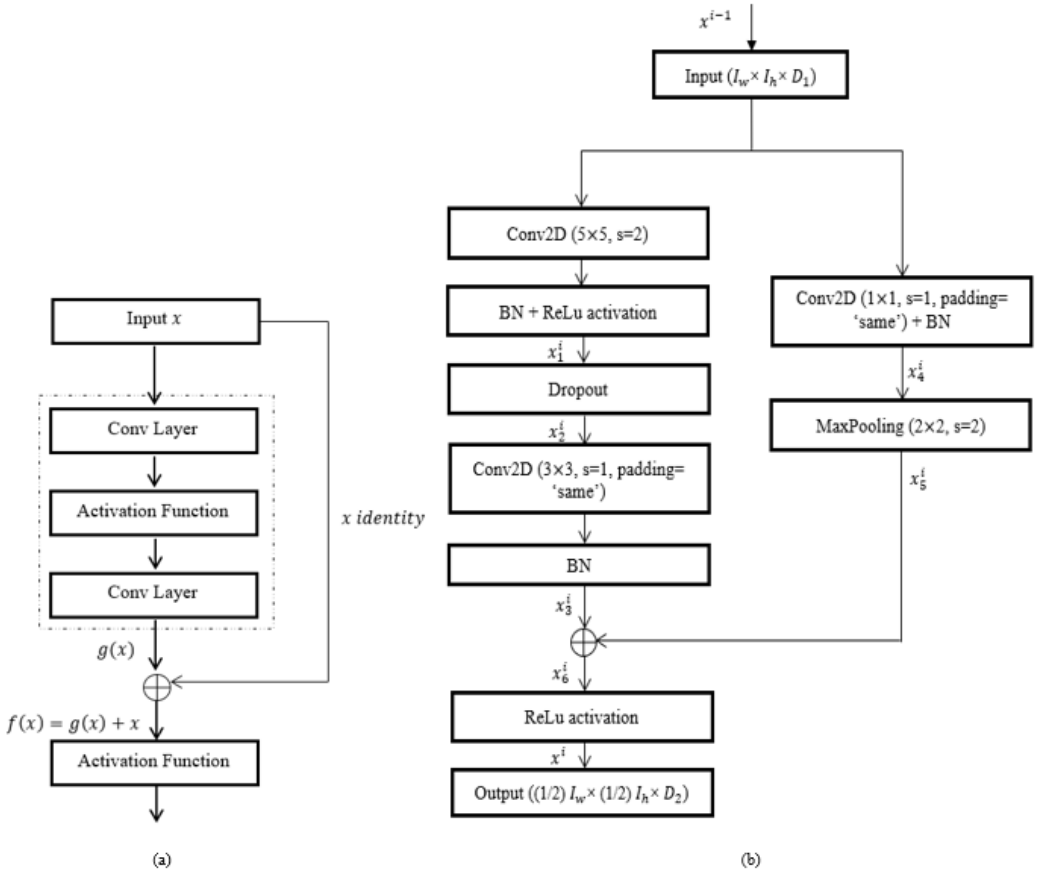

Fig. 1: Original and modified residual block: (a) Original residual block; (b) The proposed MRC block. Source: created by the authors.

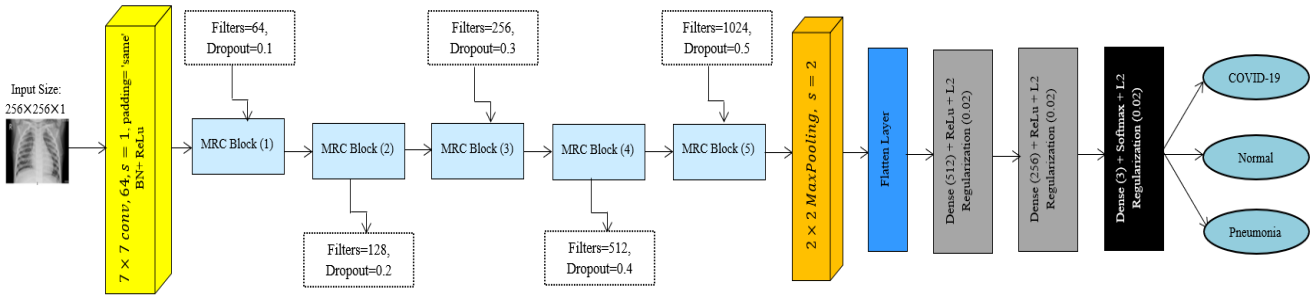

Fig. 2: Architecture of the proposed COVID-MRSNet model. Source: created by the authors.

Table 1: Layer configuration details of the proposed COVID-MRSNet model. Source: created by the authors.

Layer Name	Output shape	Activation	Parameters
Input Image	(256,256,1)	-	0
Conv2D (f=64, k=(7,7), s=1, padding='same')	(256,256,64)	ReLU	3,200
Batch Normalization	(256,256,64)	-	256
MRC Block (1), Dropout=0.1	(128,128,64)	ReLU	144,064
MRC Block (2), Dropout=0.2	(64,64,128)	ReLU	362,368
MRC Block (3), Dropout=0.3	(32,32,256)	ReLU	1,445,632
MRC Block (4), Dropout=0.4	(16,16,512)	ReLU	5,774,848
MRC Block (5), Dropout=0.5	(8,8,1024)	ReLU	23,084,032
AveragePooling2D (pool size=(2,2), s=2)	(4,4,1024)	-	0
Flatten	16,384	-	0
Dense (units=512, kernel-regularizer=$L2$)	512	ReLU	8,389,120
Dense (256, kernel-regularizer=$L2$)	256	ReLU	131,328
Dense (3, kernel-regularizer=$L2$)	3	Softmax	771
Total Parameters: 39,335,619			
Trainable Parameters: 39,323,715			
Non-trainable Parameters: 11,904			

Table 2: Dataset details used in training, validation and testing. Source: [35]

Disease	Total	Training	Validation	Testing
Normal images	1341	965	107	269
Viral-Pneumonia images	1345	969	107	269
Coronavirus-infection images	219	158	18	43

Table 3: Model parameters. Source: created by the authors.

Parameter	Value
Epochs	50
Optimizer	Adam
Learning rate	0.00001
Batch size	32
Cost function	Categorical Cross Entropy

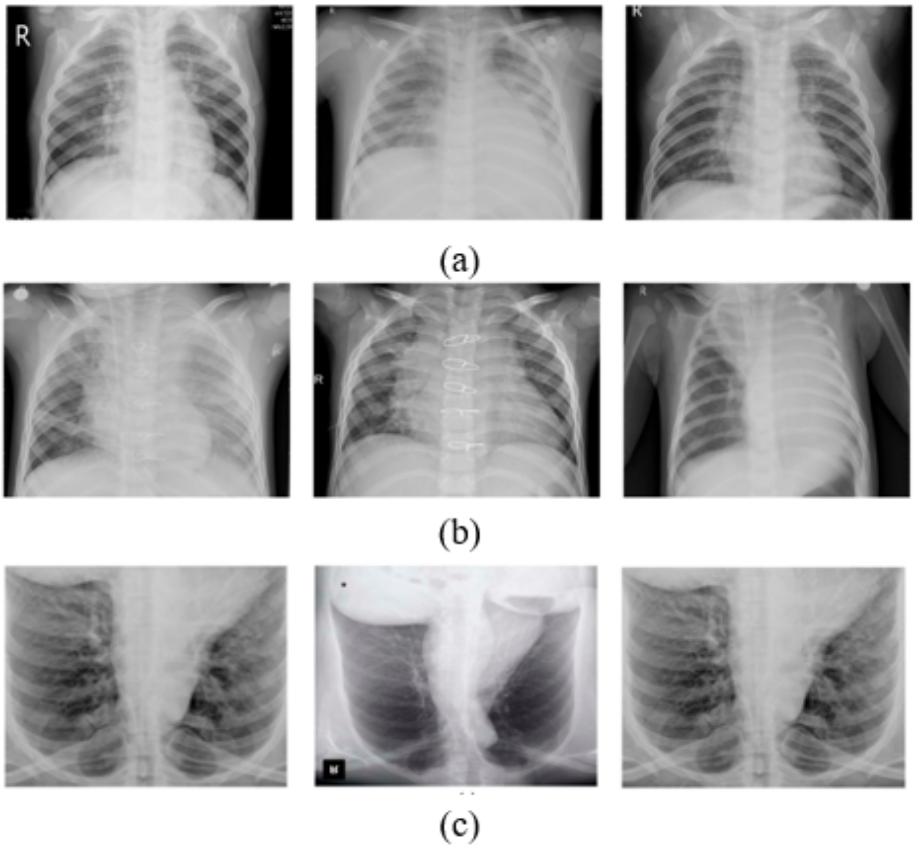

Fig. 3: Sample of the collected dataset: (a) Coronavirus-infection images, (b) viral pneumonia, and (d) normal images. Source: [35]

Table 4: Numerical results of our model at fold 1 across all categories. Source: created by the authors.

Class	Precision	F1-score	Recall
COVID-19	0.9767	0.9767	0.9767
Normal	0.9924	0.9793	0.9665
Viral Pneumonia	0.9674	0.9798	0.9926

Table 5: Numerical results of our model at fold 2 across all categories. Source: created by the authors.

Class	Precision	F1-score	Recall
COVID-19	1	1	1
Normal	0.9888	0.9869	0.9851
Viral Pneumonia	0.9852	0.9870	0.9888

Table 6: Numerical results of our model at fold 3 across all categories. Source: created by the authors.

Class	Precision	F1-score	Recall
COVID-19	1	1	1
Normal	0.9887	0.9851	0.9814
Viral Pneumonia	0.9816	0.9852	0.9889

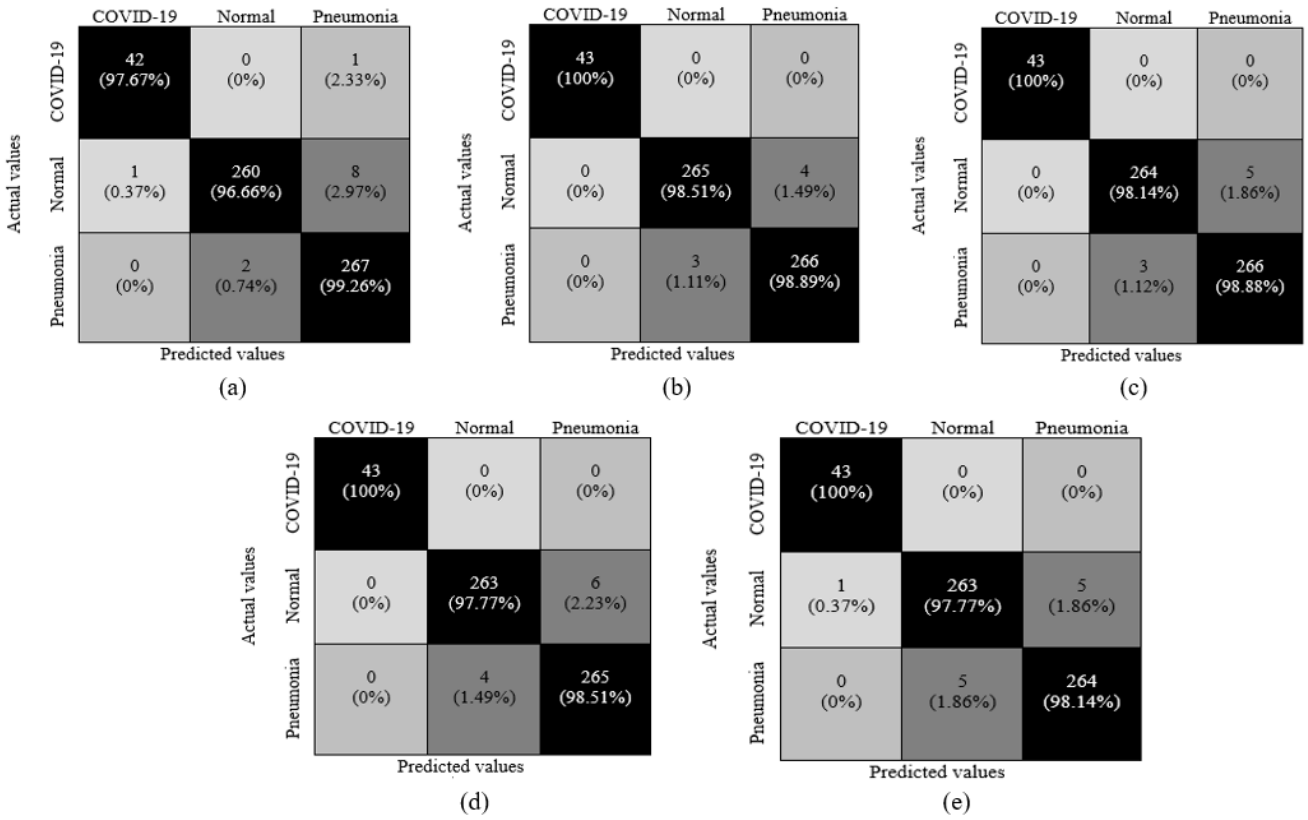

Fig. 4: Confusion matrices produced by the proposed COVID-MRSNet for the three-class classification: (a) Fold 1, (b) Fold 2, (c) Fold 3, (d) Fold 4, and (e) Fold 5. Source: created by the authors.

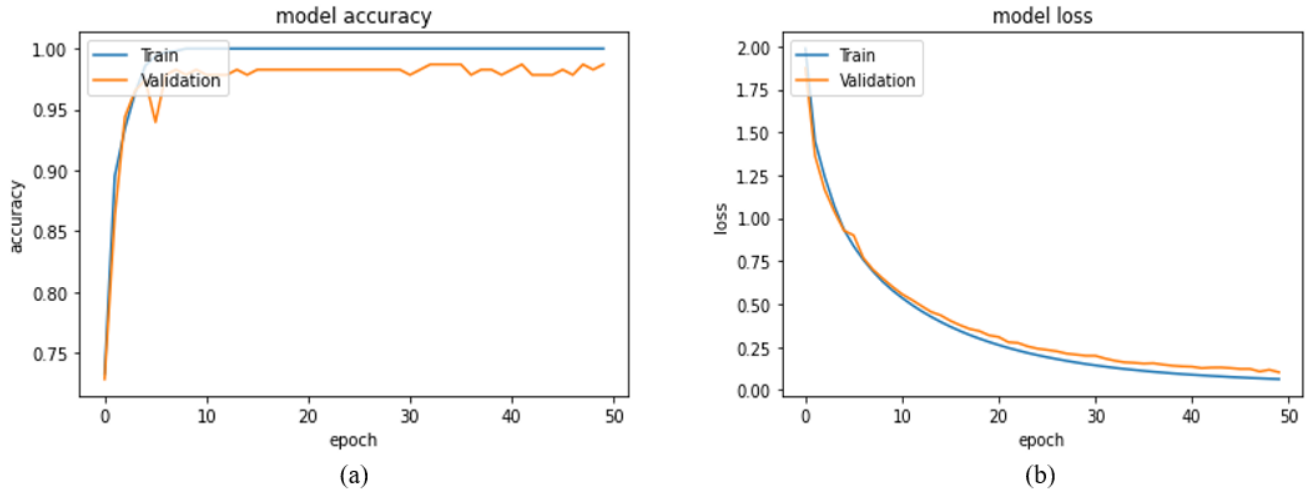

Fig. 5: Performance analysis of COVID-MRSNet: (a) training and validation accuracies at fold-1; (b) training and validation losses at fold-1. Source: created by the authors.

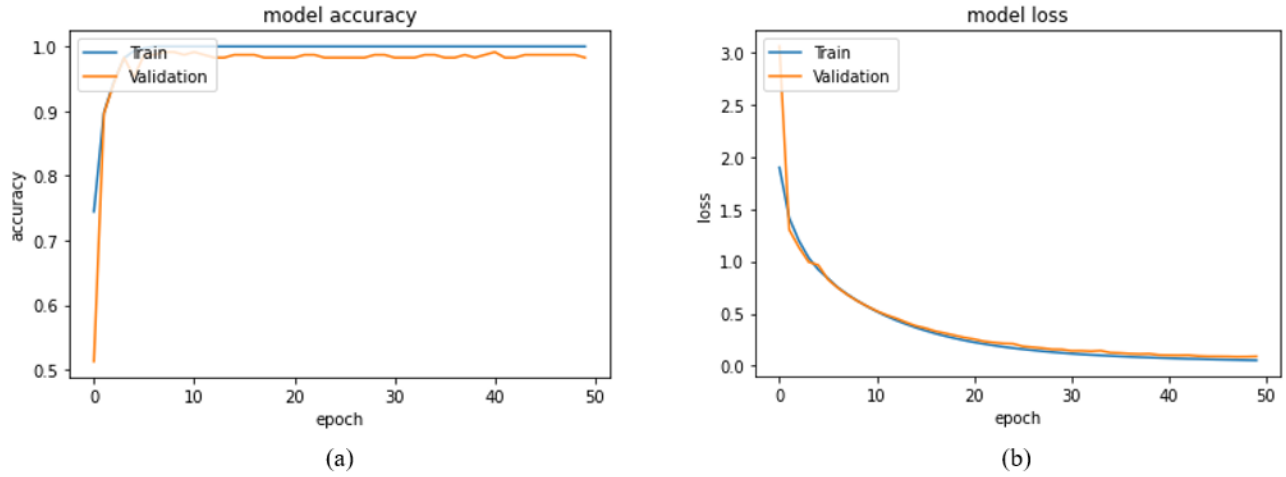

Fig. 6: Performance analysis of COVID-MRSNet: (a) training and validation accuracies at fold-2; (b) training and validation losses at fold-2. Source: created by the authors.

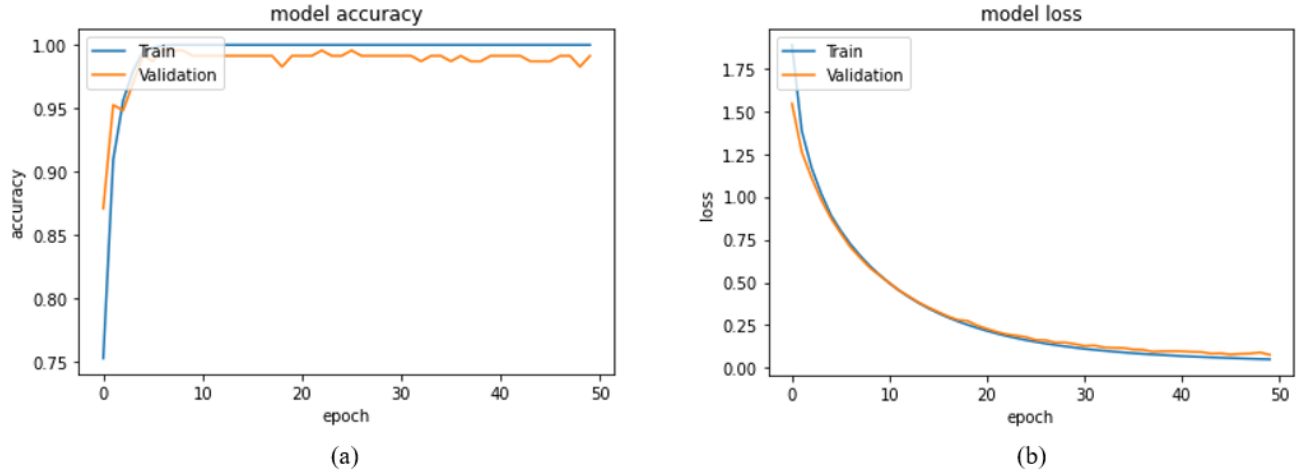

Fig. 7: Performance analysis of COVID-MRSNet: (a) training and validation accuracies at fold-3; (b) training and validation losses at fold-3. Source: created by the authors.

Table 7: Numerical results of our model at fold 4 across all categories. Source: created by the authors.

Class	Precision	F1-score	Recall
COVID-19	1	1	1
Normal	0.9850	0.9813	0.9777
Viral	0.9778	0.9815	0.9851

Table 8: Numerical results of our model at fold 5 across all categories. Source: created by the authors.

Class	Precision	F1-score	Recall
COVID-19	0.9772	0.9885	1
Normal	0.9813	0.9795	0.9777
Viral Pneumonia	0.9814	0.9814	0.9814

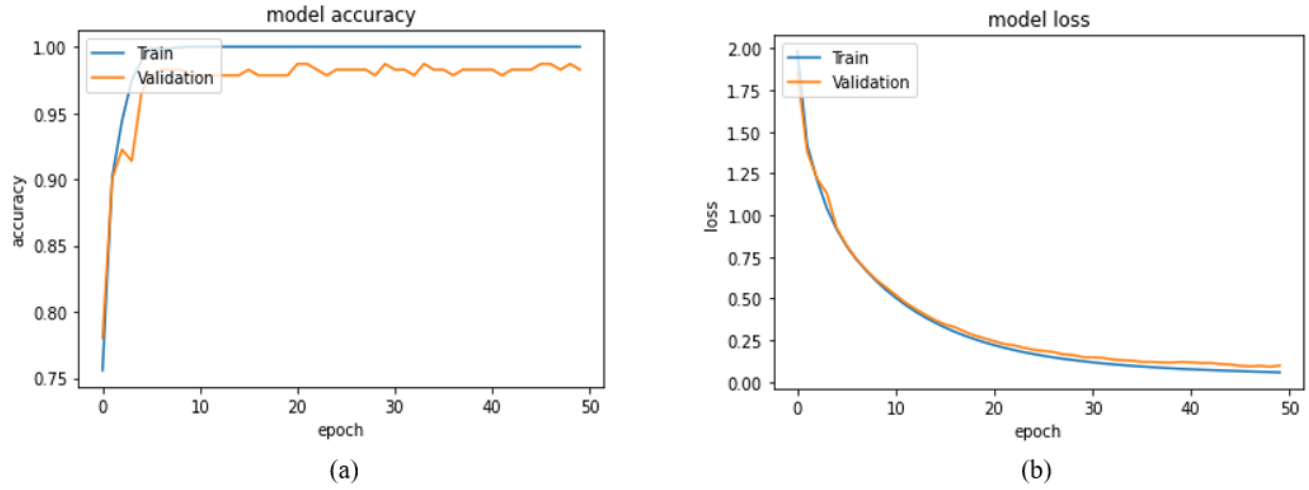

Fig. 8: Performance analysis of COVID-MRSNet: (a) training and validation accuracies at fold-4; (b) training and validation losses at fold-4. Source: created by the authors.

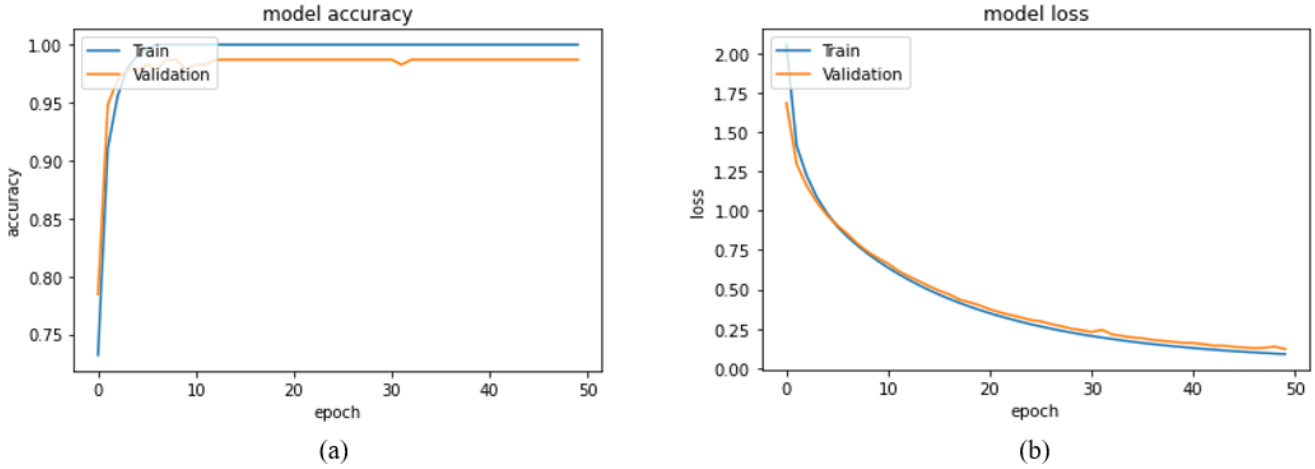

Fig. 9: Performance analysis of COVID-MRSNet: (a) training and validation accuracies at fold-5; (b) training and validation losses at fold-5. Source: created by the authors.

Table 9: Numerical results of our model using fivefold cross-validation. Source: created by the authors.

K-fold	Accuracy (%)	Precision (%)	F1-score (%)	Recall (%)
Fold 1	97.94	97.88	97.86	97.86
Fold 2	98.79	99.13	99.13	99.13
Fold 3	98.62	99.01	99.00	99.00
Fold 4	98.28	98.76	98.76	98.76
Fold 5	98.10	98.00	98.31	98.63
Mean	98.34	98.55	98.61	98.67

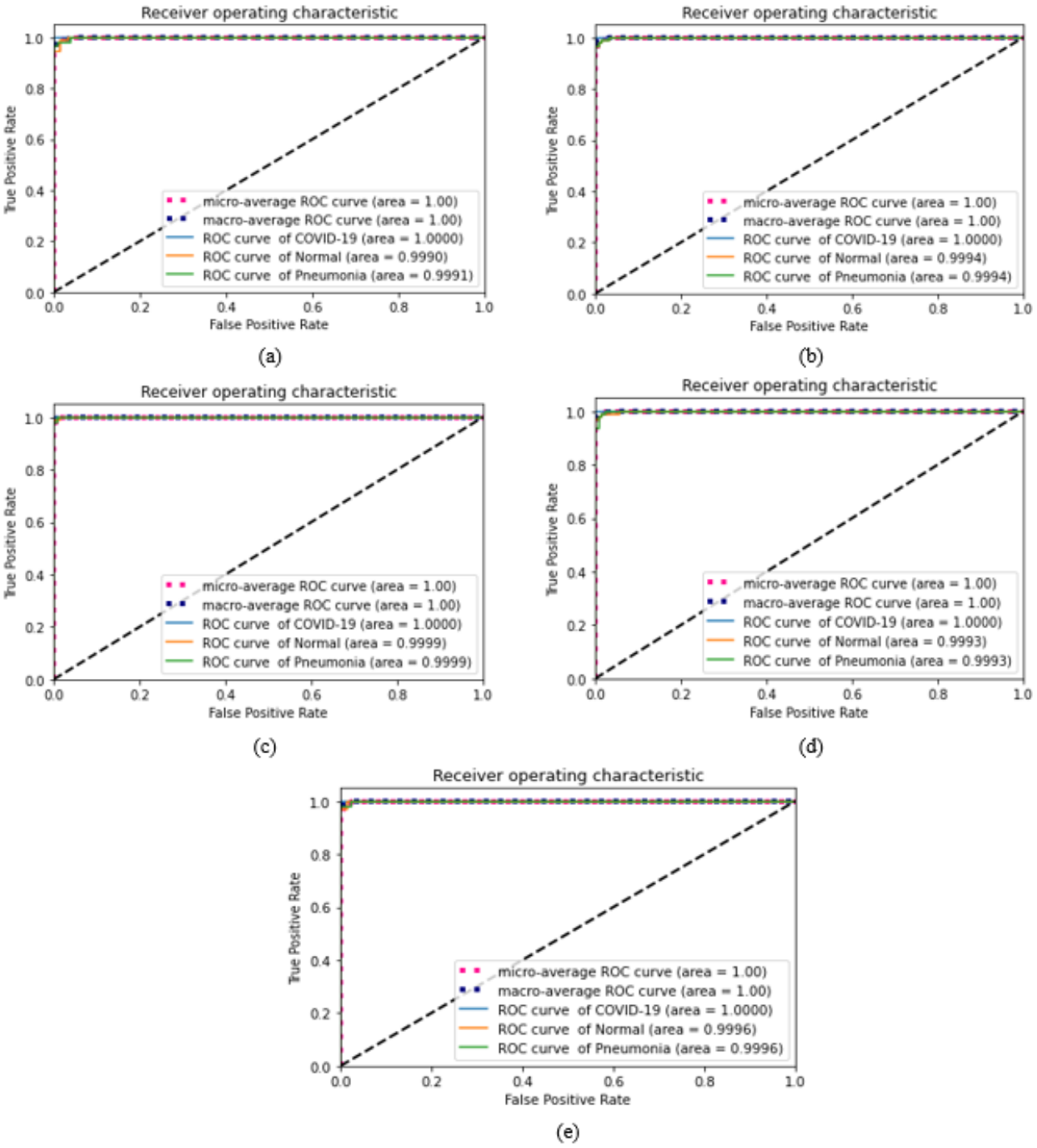

Fig. 10: ROC curve of the introduced framework for COVID-19, Normal and Viral Pneumonia on the test data set: (a) ROC-Curves at fold-1; (b) ROC-Curves at fold-2; (c) ROC-Curves at fold-3; (d) ROC-Curves at fold-4; (e) ROC-Curves at fold-5. Source: created by the authors.

Table 10: Evaluation of the introduced framework with previous studies for three-class categorization (reported in %). Source: created by the authors.

Model	Dataset	Accuracy	Precision	F1-score	Recall
Xception net	6432 chest x-rays	97.97	97.0	86.0	78.0
CoroNet	1300 x-rays	95.00	95.00	95.60	96.90
Covid-net	13975 x-rays	93.3	-	-	91
CXGNet	248 x-rays	97.05	94.44	95.38	96.96
ECONET	15493 x-rays	96.07	96.28	96.26	96.26
TL-CNN	1427 x-rays	90.39	89.46	93.38	92.39
CoroDet	400 x-rays	94.2	94.04	91.32	92.5
DarkNet	1125 x-rays	86.37	88.84	90.29	91.39
CapsNet	3150 x-rays	84.22	-	-	-
EfficientNet	404 x-rays	96.70	96.69	97.11	97.54
The Proposed COVID-MRSNet	2905 x-rays	98.34	98.55	98.61	98.67