Research Article

Diagnosis of COVID-19 from X-rays using combined CNN-RNN architecture with transfer learning

Md. Milon Islam [a],[*], Md. Zabirul Islam [a], Amanullah Asraf [a], Mabrook S. Al-Rakhami [b], Weiping Ding [c],[**], Ali Hassan Sodhro [d],[e]

[a] Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh
[b] Research Chair of Pervasive and Mobile Computing, Information Systems Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[c] School of Information Science and Technology, Nantong University, Nantong 226019, China
[d] Department of Computer Science, Kristianstad University, SE-29188 Kristianstad, Sweden
[e] Shenzhen Institutes of Advanced technology, Chinese Academy of Sciences, Shenzhen, China

ARTICLE INFO

ABSTRACT

Combating the COVID-19 pandemic has emerged as one of the most promising issues in global healthcare. Accurate and fast diagnosis of COVID-19 cases is required for the right medical treatment to control this pandemic. Chest radiography imaging techniques are more effective than the reverse-transcription polymerase chain reaction (RT-PCR) method in detecting coronavirus. Due to the limited availability of medical images, transfer learning is better suited to classify patterns in medical images. This paper presents a combined architecture of convolutional neural network (CNN) and recurrent neural network (RNN) to diagnose COVID-19 patients from chest X-rays. The deep transfer techniques used in this experiment are VGG19, DenseNet121, InceptionV3, and Inception-ResNetV2, where CNN is used to extract complex features from samples and classify them using RNN. In our experiments, the VGG19-RNN architecture outperformed all other networks in terms of accuracy. Finally, decision-making regions of images were visualized using gradient-weighted class activation mapping (Grad-CAM). The system achieved promising results compared to other existing systems and might be validated in the future when more samples would be available. The experiment demonstrated a good alternative method to diagnose COVID-19 for medical staff.

All the data used during the study are openly available from the Mendeley data repository at https://data.mendeley.com/datasets/mxc6vb7svm. For further research, we have made the source code publicly available at https://github.com/Asraf047/COVID19-CNN-RNN.

1. Introduction

The COVID-19 outbreak has spread rapidly due to person-to-person transmission and created a devastating impact on global health. So far, COVID-19 has infected the world with over 665,336,000 infections and close to 6,698,000 deaths [1]. Healthcare systems have been broken down in all countries due to the limited number of intensive care units (ICUs). Coronavirus-infected patients with serious conditions are admitted into ICUs. To control this pandemic, many national governments have presented 'lockdown' to ensure 'social distancing' and 'isolation' among the people to reduce person-to-person transmission [2]. The coronavirus symptoms may vary from cold to fever, acute respiratory illness, and shortage of breath [3]. The most crucial step is to diagnose COVID-19 at an early stage and isolated the infected people from others. RT-PCR is considered a key indicator to diagnose COVID-19

cases reported by the government of China [4]. However, it is a time-consuming method with a high false negatives rate [5]. In many cases, the coronavirus affected may not be identified correctly for the low sensitivity.

To combat this pandemic, a lot of interest has been found in the role of medical imaging modalities [6]. Chest radiographs such as chest X-ray and computed tomography (CT) are suitable for the detection of COVID-19 due to the high sensitivity that is already explored as a standard diagnosis system for pneumonia disease [7]. CT scan is more accurate than a chest X-ray to diagnose pneumonia but still chest X-ray is effective due to cheaper, quicker, and fewer radiation systems [8]. Deep learning [9–11] is widely used in the medical field for the analysis of complex medical images. Deep learning techniques rely on automated extracted features instead of manual handcrafted features.

---

* Corresponding author.
** Second corresponding author.
E-mail addresses: milonislam@cse.kuet.ac.bd (Md.M. Islam), zabir.kuet.cse@gmail.com (Md.Z. Islam), amanullahoasraf@gmail.com (A. Asraf), malrakhami@ksu.edu.sa (M.S. Al-Rakhami), dwp9988@163.com (W. Ding), ali.hassan_sodhro@hkr.se (A.H. Sodhro).

We proposed a combination of CNN and RNN frameworks to identify coronavirus cases from chest X-rays in this paper. The RNN network is capable of handling long-term dependencies using internal memories. In the case of fully connected networks, nodes between layers are connectionless and process only one input but in RNN, nodes are connected from a directed graph that processes an input with a specific order [12–14]. We comparatively used four pre-trained CNN models namely VGG19, DenseNet121, InceptionV3, and Inception-ResNetV2 with RNN to find out the best CNN-RNN architecture within the limitations of the datasets. In this system, we first used the pre-trained CNN to extract significant features from images. Then, we applied the RNN classifier to identify COVID-19 cases using the extracted features. The contributions of this paper are summarized in the following.

(i) We developed and trained a combined four CNN-RNN architectures to classify coronavirus infection from others.
(ii) To detect coronavirus cases, a total of 6396 X-ray samples are used as a dataset from several sources.
(iii) An exhaustive experimental analysis is ensured to measure the performance of each architecture in terms of area under the receiver operating characteristics (ROC) curve (AUC), accuracy, precision, recall, F1-score, and confusion matrix and also applied Grad-CAM to visualize the infected region of X-rays.

The paper is organized as follows. A brief review of related works is presented in Section 2. The methods and materials including dataset collection, the development of combined networks, and performance evaluation metrics are described in Section 3. Extensive result analysis with relative discussions is illustrated in Section 4. Finally, the conclusion of the paper is drawn in Section 5.

2. Related works

Because of the COVID-19 pandemic, many efforts have been explored to develop a system for the diagnosis of COVID-19 using artificial intelligence techniques such as machine learning [15], and deep learning [16]. In this section, a detailed description is provided of the recently developed systems to diagnose COVID-19 cases.

Luz et al. [17] introduced an extended EfficientNet model based on convolutional network architecture to analyze lung conditions using X-ray images. The model used 183 samples of COVID-19 and achieved 93.9% accuracy and 80% sensitivity for coronavirus classification. Rahimzadeh and Attar [18] presented a concatenated Xception and ResNet50V2 network to find out the infected region of COVID-19 patients from chest X-rays. The network trained in eight phases and used 633 samples in each phase including 180 samples of COVID-19. The network obtained 99.56% accuracy and 80.53% recall to detect coronavirus infection. Minaee et al. [19] illustrated a deep transfer learning architecture utilizing 71 COVID-19 samples to identify infected parts from other lung diseases. The architecture obtained an overall 97.5% sensitivity and 90% specificity to differentiate coronavirus cases. Punn and Agarwal [20] demonstrated a deep neural network to identify coronavirus symptoms. The scheme used 108 COVID-19 cases and obtained an average of 97% accuracy. Khan et al. [21] introduced a deep CNN to diagnose coronavirus disease from 284 COVID-19 samples. The proposed framework found an accuracy of 89.5%, and a precision of 97% to detect coronavirus. Wang and Wong [22] presented COVID-Net to distinguish COVID-19 cases from others using chest X-ray samples. The system achieved 92.4% accuracy by analyzing 76 samples of COVID-19. Narin et al. [23] proposed deep transfer learning with three CNN architectures and used a small dataset including 50 chest X-rays for both COVID-19 and normal cases to detect coronavirus infection. The ResNet50 showed high performance with 98.6% accuracy, 96% recall, and 100% specificity among other networks.

Hemdan et al. [24] developed a COVIDX-Net framework including seven pre-trained CNN to detect coronavirus infection from X-ray samples. The dataset consisted of 25 samples of COVID-19 cases and 25 samples of normal cases. The framework obtained high performance for VGG19 with a 0.89 F1 score. Apostolopoulos and Mpesiana [25] presented a transfer learning scheme for the detection of coronavirus infection. The VGG19 obtained high performance among others with an accuracy of 93.48%, specificity of 92.85%, and sensitivity of 98.75%. Horry et al. [26] illustrated a deep transfer learning-based system and achieved the highest result for VGG19 with 83% recall and 83% precision for the diagnosis of COVID-19. Loey et al. [27] proposed a deep transfer learning approach with three pre-trained CNN networks to diagnose coronavirus disease. The dataset includes 69 COVID-19 samples, 79 pneumonia bacterial samples, 79 pneumonia virus samples, and 79 normal samples. The GoogleNet achieved an accuracy of 80.6% in the four cases scenario. Kumar and Kumari [28] used a transfer learning-based system using nine pre-trained CNNs combined with a support vector machine (SVM) to classify coronavirus-infected patients. The ResNet50-SVM showed the best performance among other models with an accuracy of 95.38%. Bukhari et al. [29] proposed a transfer learning technique for the detection of COVID-19 from X-ray samples. The system used 89 samples of COVID-19 and obtained 98.18% accuracy with a 98.19% F1-score. Abbas et al. [30] introduced a DeTraC architecture to detect coronavirus from 105 samples of COVID-19. The architecture achieved 95.12% accuracy, 91% sensitivity, 91.87% specificity, and 93.36% precision to diagnose coronavirus infection. Islam et al. [31] applied a combined CNN and LSTM architecture to classify coronavirus cases using X-ray images. The scheme applied 421 samples including 141 COVID-19 cases and achieved an accuracy of 97%, specificity of 91%, and sensitivity of 93%.

Faisal et al. [32] developed two- and three-classifier diagnosis frameworks for classifying COVID-19 patients using transfer-learning approaches that obtained an accuracy of 99.5% and 98.3% for binary and multi-class classification. Dey et al. [33] used an ML-based system with a sequence of tasks ranging from image pre-processing for the classification of COVID-19 with higher than 90% accuracy. Wang et al. [34] introduced He presented a 5G-enabled auxiliary diagnosis architecture based on federated learning for many organizations and centralized cloud cooperation to facilitate the sharing of high-generalization diagnosis tools. Singh et al. [35] illustrated a pipeline using a Hybrid Social Group Optimization algorithm to classify COVID-19 patients from chest X-rays with 99.65% accuracy. Gumaei et al. [36] developed a regression method for COVID-19 confirmed cases prediction to make future forecasting of the ongoing pandemic.

3. Methods and materials

Though some of the existing systems showed promising results, the COVID-19 dataset was quite small [19,24,27], and also the variable quality of these datasets was not addressed. It also noticed that the used dataset in those experiments was quite unbalanced which could lead to the over-classification of the majority class at the expense of the under-classification of the minority class [21,22]. On the contrary, COVID-19 images were highly inconsistent as they were retrieved from different regions of the world whereas pneumonia and normal images were uniform as well as highly curated in previous studies. Here, the COVID-19 dataset contained most adult patients, and the pneumonia dataset used mostly young patients. These discrepancies were mostly ignored in the existing systems [29]. Therefore, our proposed system used a balanced dataset with adult and young patients' images to learn the actual features of the disease. The proposed system contains several steps to diagnose COVID-19 infection as shown in Fig. 1. Firstly, in the preprocessing pipeline, chest X-ray samples were resized, shuffled, and normalized to figure out the actual features and reduce the unwanted noise from the images. Afterward, the dataset was partitioned into training and testing sets. Using the training dataset, we applied four pre-trained CNN architectures combined with the RNN classifier. The accuracy and loss of training datasets were obtained after each epoch and using a five-fold cross-validation technique, the validation loss and accuracy were found. The performance of the overall system was measured with a confusion matrix, accuracy, precision, recall, AUC, and F1-score metrics.
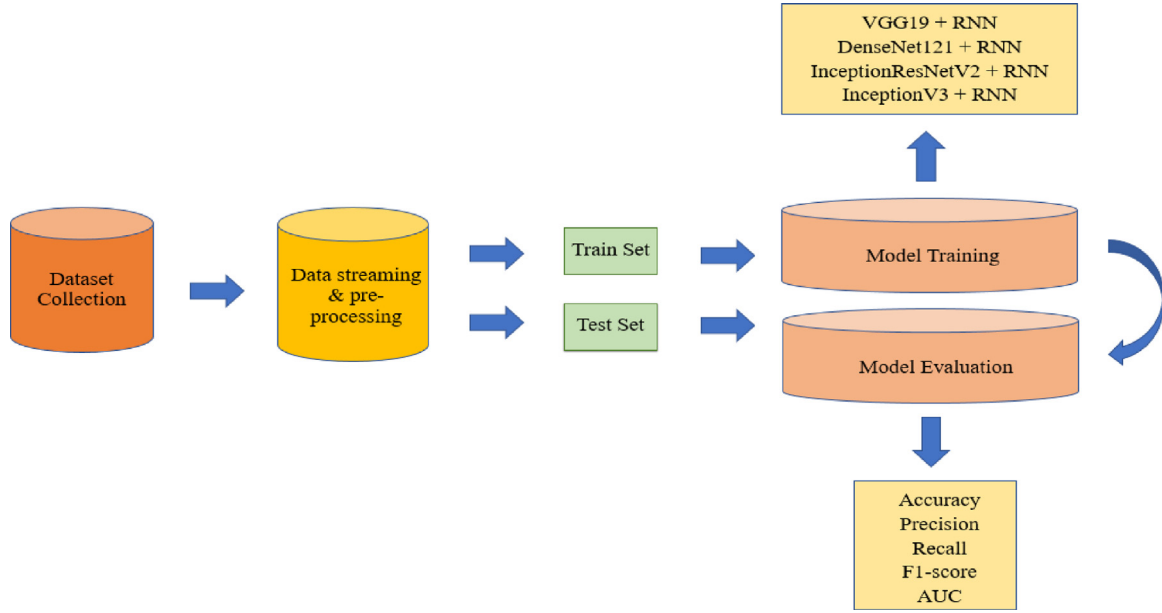
Fig. 1. The overall system architecture of the COVID-19 diagnosis framework.

Table 1
The dataset of our proposed CNN-RNN model.

Images	COVID-19	Pneumonia	Normal	Total
Training	1850	1851	1850	5551
Testing	463	462	463	1388
Total	2313	2312	2313	6939

Table 2
Characteristics of four pre-trained CNN architectures.

Network	Depth	Parameters (10^6)
VGG19	26	143.67
DenseNet121	121	8.06
InceptionV3	159	23.85
Inception-ResNetV2	572	55.87

3.1. Experimental dataset

In this paper, X-ray samples of COVID-19 were retrieved from seven different sources of the unavailability of a large specific dataset. Firstly, a total of 1401 samples of COVID-19 were collected using the GitHub repository [37,38], the Radiopaedia [39], Italian Society of Radiology (SIRM) [40], Figshare data repository websites [41,42]. Then, 912 augmented images were also collected from Mendeley instead of using data augmentation techniques explicitly [43]. Finally, 2313 samples of normal and pneumonia cases were obtained from Kaggle [44,45]. A total of 6939 samples were used in the experiment, where 2313 samples were used for each case. The total dataset was divided into 80%–20% for training and testing sets where 1850 samples of COVID-19, 1851 samples of pneumonia, and 1850 samples of normal cases were used for training including all augmented images shown in Table 1. The remaining 463 samples of COVID-19, 462 samples of pneumonia, and 463 samples of normal cases were used for the testing including only original images; no augmented images were used here. Pixel normalization was applied to images in data preprocessing step.

3.2. Development of combined network

3.2.1. Deep transfer learning with CNN

Transfer Learning [46] is an approach where information extracted by one domain is transferred to another related domain. It is applied when the dataset is not sufficient to train the parameters of any network. In this part, four pre-trained CNNs are described to accomplish the proposed CNN-RNN architecture as follows. In addition, the characteristics of four pre-trained CNN architectures are shown in Table 2.

(i) **VGG19:** VGG19 [47] is a version of the visual geometry group network (VGG) developed by Karen Simonyan and Andrew

Zisserman based on deep network architecture. It has 19 layers in total including 16 convolutional layers with three fully-connected layers to perform on the ImageNet dataset [48]. VGG19 used a 3×3 convolutional filter and a stride of 1 that was followed by multiple non-linear layers. Max-pooling is applied in VGG19 to reduce the volume size of the image and achieved high accuracy in image classification.

(ii) **DenseNet121:** Dense Convolutional Network (DenseNet) [49] uses dense connections instead of direct connections among the hidden layers developed by Huang et al. In DenseNet architecture, each layer is connected to the next layer to transfer the information among the network. The feature maps are transmitted directly to all subsequent layers and use only a few parameters for training. The overfitting of a model is reduced by dense connections for small datasets. DenseNet121 has 121 layers, loaded with weights from the ImageNet dataset.

(iii) **InceptionV3:** InceptionV3 [50] is used to improve computing resources by increasing the depth and width of the network. It has 48 layers with skipped connections to use a building block and is trained on million images including 1000 categories. The inception module is repeated with max-pooling to reduce dimensionality.

(iv) **Inception-ResNetV2:** Inception-ResnetV2 [51] network is a combination of inception structure with residual connections including 164 deep layers. It has multiple-sized convolution filters trained on millions of images and avoids the degradation problem.

3.2.2. Recurrent neural network

A recurrent neural network [52] is an extended feedforward neural network with one or more feedback loops designed for processing sequential data. Given, an input sequence $(x_1, \ldots, x_t)$, an RNN generates
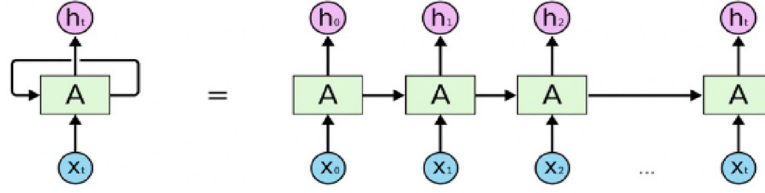
Fig. 2. The structure of recurrent neural networks.

an output sequence of $(y_1, \ldots, y_t)$ by using the following formulas, and the RNN structure is shown in Fig. 2.

$$h_t = sigm(W^{hx}X_t + W^{hh}h_{t-1}) \tag{1}$$

$$y_t = W^{yh}h_t \tag{2}$$

RNN is used whenever the input–output relationship is found based on time and capacity to handle long-term dependencies [53]. The strategy of modeling sequence is to feed the input sequence to a fixed-sized vector using an RNN, and then to map the vector to a softmax layer. Unfortunately, a problem occurs in RNN when the gradient vector is increasing and decreasing exponentially for long sequences. This vanishing gradient and exploding problem [54] create difficulties to learn long-range relationships from the sequences of the RNN architecture. However, Long Short-Term Memory (LSTM) [55] is capable to solve such a long-distance dependencies problem successfully. The main difference from RNN is that LSTM added a separate memory cell state to store long-term states and updates or exposes them whenever necessary. The LSTM consists of three gates: input gate, forget gate, and output gate where i_t denotes input gate, f_t denotes forget gate, O_t denotes output gate, $\tilde{C}$ cell input activation vector, c_t refers to the memory cell state, and h_t refers to the hidden state at each time step t.

The transition representations of LSTM are as follows.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + V_i c_{t-1}) \tag{3}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + V_f c_{t-1}) \tag{4}$$

$$O_t = \sigma(W_o x_t + U_o h_{t-1} + V_o c_{t-1}) \tag{5}$$

$$\tilde{C} = \tanh(W_c x_t + U_c h_{t-1}) \tag{6}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{C} \tag{7}$$

$$h_t = O_t \odot \tanh(c_t) \tag{8}$$

where x_t refers to current input, σ refers to the sigmoid function and $\odot$ refers to element-wise multiplication.

3.2.3. Development of CNN-RNN hybrid network

A combined method using CNN and RNN was developed for the diagnosis of COVID-19 using three types of X-ray samples in this paper. The complex features were extracted from $224 \times 224 \times 3$ sized samples using VGG19, DeneNet121, InceptionV3, and Inception-ResNetV3. The extracted features were fed to the single-layered RNN classifier i.e. the output is produced by passing it through a single hidden state to differentiate COVID-19, pneumonia, and normal cases. The dimensionality of feature maps of pre-trained CNN and how CNN is connected to RNN were shown in Table 3.

The CNN-RNN network for COVID-19 classification is shown in Fig. 3 which contains the following steps.

Step 1: Use different pre-trained CNN models to extract essential features from X-ray images.
Step 2: Reshape the feature map into the sequence.
Step 3: Set the feature map as the input of a single-layered RNN.
Step 4: Apply a softmax classifier to classify COVID-19 X-ray images.
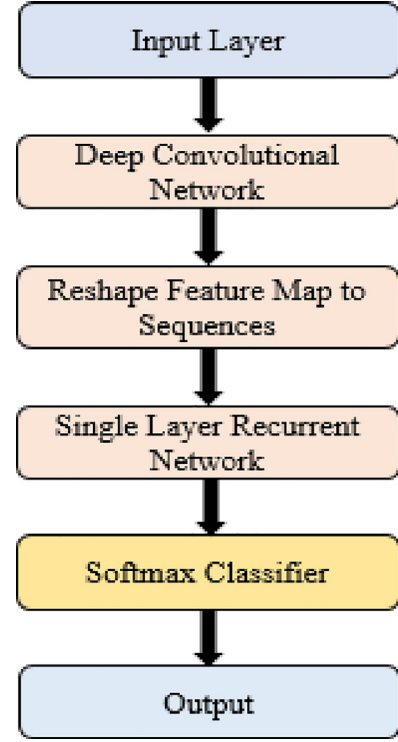

Fig. 3. The workflow of the CNN-RNN architecture for COVID-19 diagnosis.

In transfer learning, the activations of convolutional layers are the same as in original architectures. Finally, the fully connected layers were activated using Rectified Linear Unit (ReLU) [56] and the Dropout layer [57] was used in RNN layers to prevent overfitting [58] of the models. All the layers of pre-trained CNN were frozen during training except RNN and fully connected layers. Finally, the CNN-RNN architectures were trained with RMSprop [59] and a batch size of 32, a learning rate of 0.00001, and a total of 100 epochs were conducted for training. The samples were shuffled in batches between epochs. We used single-layer RNN combined with pre-trained CNN shown in Fig. 3. In a single-layer RNN, the output is produced by passing it through a single hidden state to capture the structure of a sequence.

Algorithm 1: CNN-RNN Algorithm

Input: Training data $D_{training}$, Testing data $D_{testing}$, learning rate n, epoch T, pre-trained models C, Recurrent Neural Network R, number of pre-trained models P_n
Output: Best CNN-RNN model
1. Preprocess $D_{training}$
for $t = 1, 2, \ldots, P_n$ **do**
    Train the model:
    Obtain feature maps O: using C[t], n, and T
    Reshape feature maps O into sequence H
    Classify the data using R based on H
    Test the model:
    Evaluate performance using $D_{testing}$ and store the results and model
Compare results among the models to identify the best model
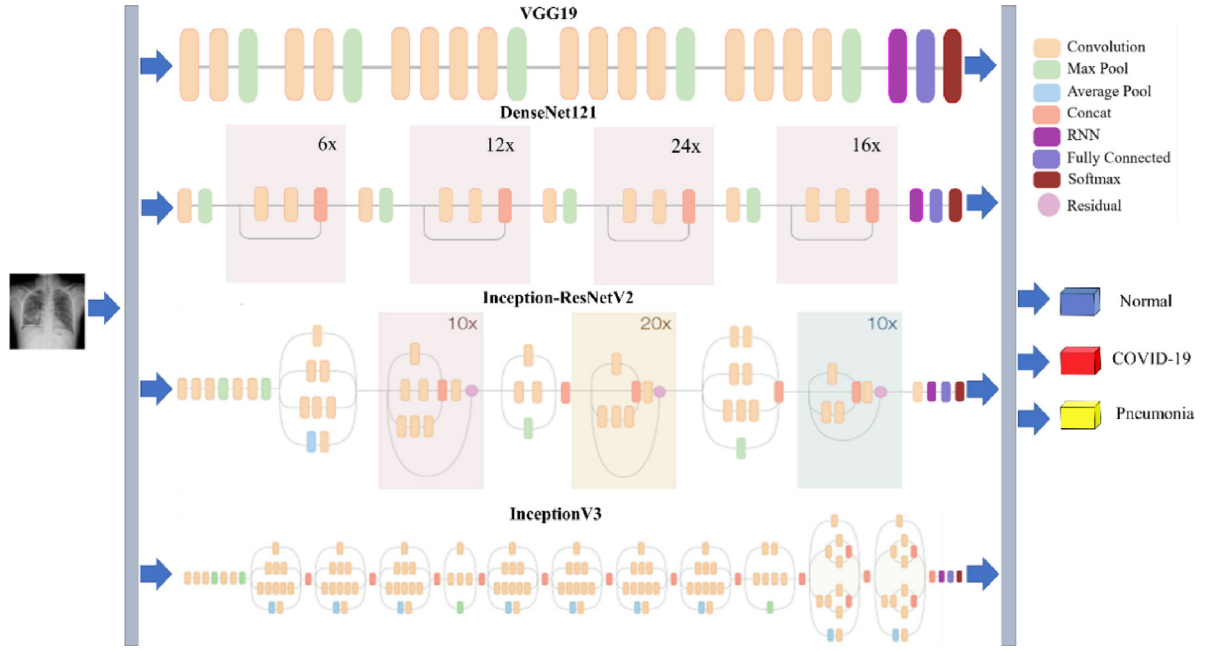   **return the best CNN-RNN model**

Fig. 4. The structure of the combined CNN-RNN architecture for COVID-19 diagnosis.

Table 3

The summary of the used architectures. (a) VGG19-RNN (b) DenseNet121-RNN (c) Inception-ResNetV2-RNN (d) InceptionV3-RNN.

(a)

Layer (type)	Input size	Output size
pretrained (VGG19)	224, 224, 3	7, 7, 512
reshape (Reshape)	7, 7, 512	49, 512
lstm (LSTM)	49, 512	49, 512
flatten (Flatten)	49, 512	25088
fc_1 (Dense)	25088	4096
fc_2 (Dense)	4096	4096
output (Dense)	4096	3

(b)

Layer (type)	Input size	Output size
pretrained (DenseNet121)	224, 224, 3	7, 7, 1024
reshape (Reshape)	7, 7, 1024	49, 1024
lstm (LSTM)	49, 1024	49, 1024
flatten (Flatten)	49, 1024	50176
fc_1 (Dense)	50176	4096
fc_2 (Dense)	4096	4096
output (Dense)	4096	3

(c)

Layer (type)	Input size	Output size
pretrained (Inception-ResNetV2)	224, 224, 3	5, 5, 1536
reshape (Reshape)	5, 5, 1536	25, 1536
lstm (LSTM)	25, 1536	25, 512
flatten (Flatten)	25, 512	12800
fc_1 (Dense)	12800	4096
fc_2 (Dense)	4096	4096
output (Dense)	4096	3

(d)

Layer (type)	Input size	Output size
pretrained (InceptionV3)	224, 224, 3	5, 5, 2048
reshape (Reshape)	5, 5, 2048	25, 2048
lstm (LSTM)	25, 2048	25, 512
flatten (Flatten)	25, 512	12800
fc_1 (Dense)	12800	4096
fc_2 (Dense)	4096	4096
output (Dense)	4096	3

We have used RNN as a sequence-to-sequence layer and taken the output sequence as input for fully-connected layers downstream in the developed system. In this architecture, Gradient clipping is used to handle the long sequence problem. In our proposed system, the order of elements of a sequence is horizontal. Basically, to develop the sequence by collecting the pixels from the images in three orders such as horizontal, vertical and spiral are used. The structure of the combined CNN-RNN is shown in Fig. 4. Algorithm 1 presented the proposed CNN-RNN technique to detect COVID-19 cases.

3.3. Evaluation criteria

The performance of the developed system is measured in terms of AUC, accuracy, precision, recall, and F1 score. The evaluation metric parameters are represented mathematically in the following. Here, correctly classified COVID-19 cases are denoted by True Positive (TP), correctly classified pneumonia or normal cases are represented by True Negative (TN), wrongly classified as COVID-19 cases are denoted by False Positive (FP), and wrongly classified as pneumonia or normal cases are depicted by False Negative (FN).

$$\text{Accuracy} = (TP + TN)/(TN + FP + TP + FN) \qquad (9)$$

$$\text{Precision} = TP/(TP + FP) \qquad (10)$$

$$\text{Recall} = TP/(TP + FN) \qquad (11)$$

$$F1 - \text{score} = (2 * \text{Precision} * \text{Recall})/(\text{Precision} + \text{Recall}) \qquad (12)$$

4. Results analysis

All the experiments were performed on a Google Colaboratory Linux server with Ubuntu 16.04 operating system using a Tesla K80 GPU graphics card and the TensorFlow/Keras framework of python language.

4.1. Results analysis

The accuracy and loss curves in the training and validation phases are shown in Fig. 5. For VGG19-RNN architecture, the highest training and validation accuracy is observed at 99.01% and 97.74% and loss is
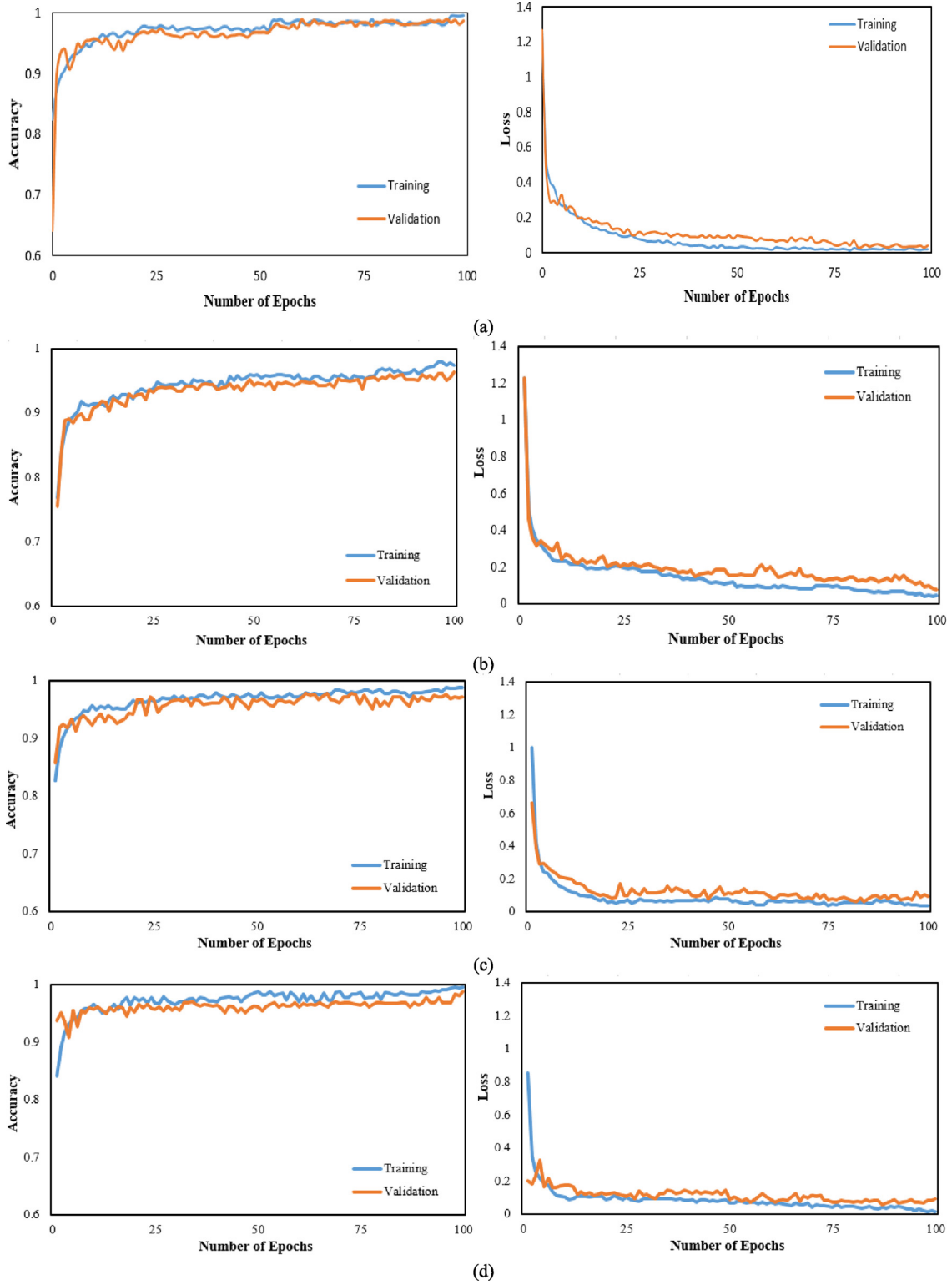
Fig. 5. Accuracy and loss curve of four CNN-RNN architectures. (a) VGG19 (b) DenseNet121(c) InceptionV3 (d) Inception-ResNetV2.

0.02 and 0.09 at epoch 100. On the contrary, the lowest training and validation accuracy is obtained 98.03% and 94.91% and loss is 0.05 and 0.26 at epoch 100 for the InceptionV3-RNN network. Analyzing the loss curve, it is seen that the loss values of VGG19-RNN decrease faster and tends to zero than other networks.

Fig. 6 demonstrates the confusion matrix of the developed architectures. Among 1388 samples, 2 samples were misclassified by

the VGG19-RNN network including only one sample for COVID-19 cases, 3 samples were misclassified by the DenseNet121-RNN network including two COVID-19 samples, 20 samples were misclassified by InceptionV3-RNN architecture consisting of three COVID-19 samples and 7 samples were misclassified by the Inception-ResNetV2-RNN network comprising of seven COVID-19 samples. Hence, it was found that VGG19-RNN architecture is superior to other networks and selected as
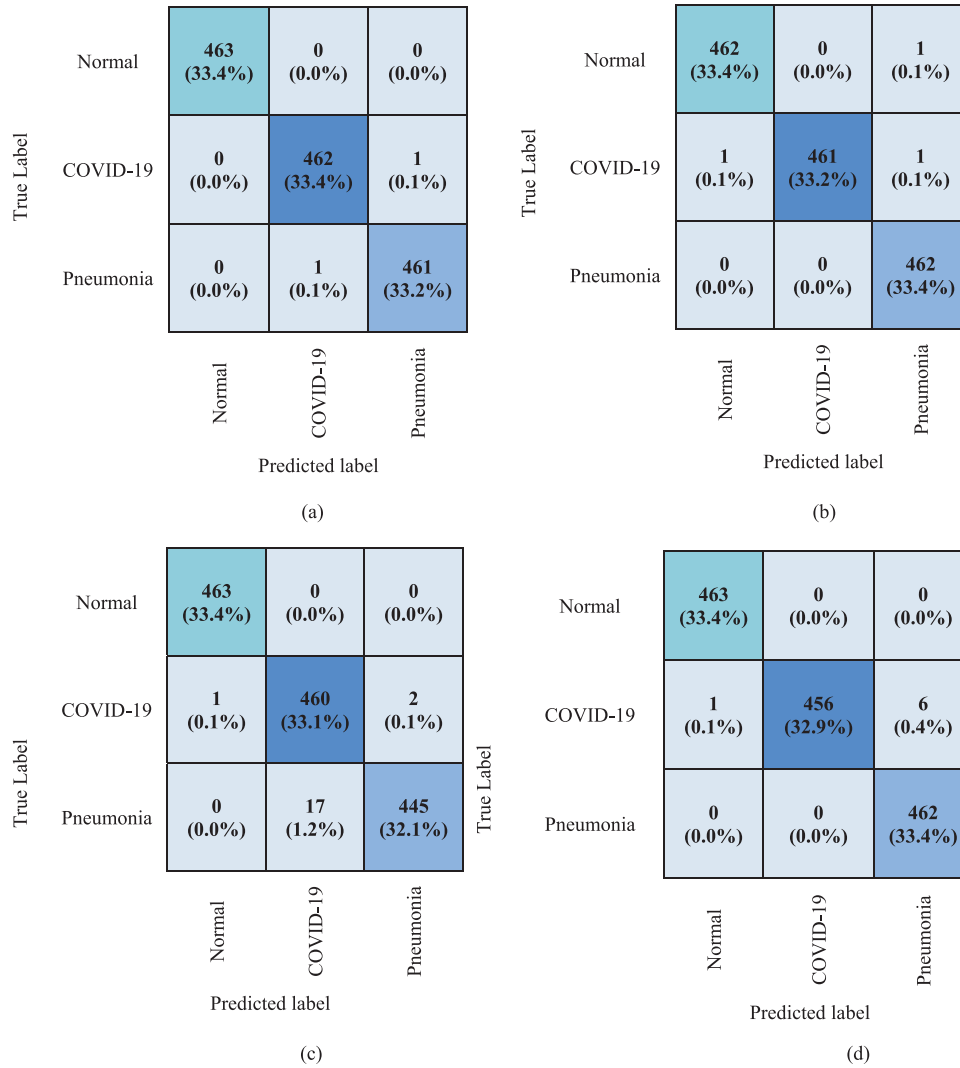
Fig. 6. Confusion matrix of the CNN-RNN architecture for COVID-19 diagnosis. (a) VGG19 (b) DenseNet121 (c) InceptionV3 (d) Inception-ResNetV2.

a main deep learning architecture with high performance. Moreover, Table 4 illustrates a comparison between the CNN-RNN network used in these experiments in terms of computational times. It is observed that VGG19-RNN achieved the highest performance and took 16722.41s for training and 129.69s for testing. In addition, it is also noticed that InceptionV3-RNN needed 16376.09s for training and 170.14s for testing.

Though InceptionV3-RNN model required less training time and more testing time than the VGG19-RNN model. It is concluded that the researcher has the choice to select the deep learning model between accuracy and computational time to use, but in the medical field, accuracy is always the main criterion. Hence, the experimental result revealed that the VGG19-RNN model outperforms other CNN-RNN architectures.

In this paper, the performance of four CNN-RNN architectures is summarized in Table 5. The best performance was found by the VGG19-RNN network with 99.86% accuracy, 99.99% AUC, 99.78% precision, 99.78% recall, and 99.78% F1-score for COVID-19 cases. On the contrary, the comparatively low performance was obtained by InceptionV3-RNN architecture with 98.56% accuracy, 99.95% AUC, 99.35% precision, 96.44% recall, and 97.87% F1-score. Besides, ROC curves were also added between TP and FP rates for all networks shown in Fig. 7. The networks can differentiate COVID-19 cases from others

Table 4
Comparative computational time CNN-RNN models.

Model	Training time (s)	Testing time (s)
VGG19	16722.41	129.69
DenseNet121	18145.67	196.02
InceptionV3	16376.09	170.14
Inception-ResnetV2	17727.26	310.63

with an AUC in the range of 99.95% to 99.99%. For better visualization and to show the differences between the classifiers Precision–Recall (PR) curve is also added shown in Fig. 8.

Finally, Grad-CAM is applied which refers to a heat map to highlight class-specific regions of chest X-rays. Fig. 9 shows the heatmaps and superimposed images of COVID-19, pneumonia, and normal cases for the VGG19-RNN network.

4.2. Discussions

In this paper, the combination of four CNNs and RNNs was used to diagnose the COVID-19 infection. The results demonstrated that VGG19-RNN is more effective to differentiate COVID-19 cases from

Table 5
Performance of the combined CNN-RNN architecture.

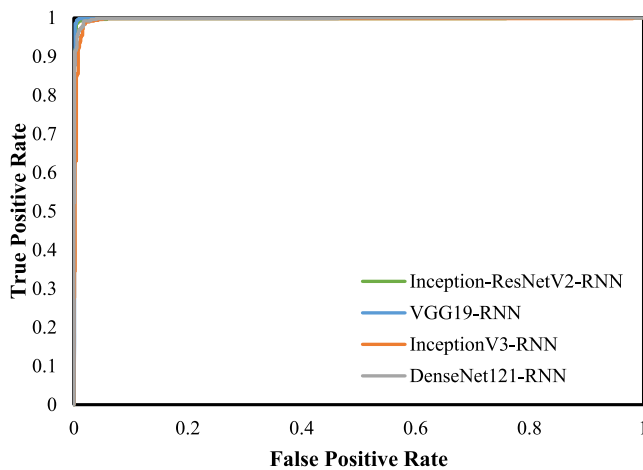| Classifier	Patient status	AUC (%)	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
VGG19-RNN	COVID-19	99.99	99.86	99.78	99.78	99.78
	Pneumonia		99.86	99.78	99.78	99.78
	Normal		99.86	100.0	100.0	100.0
DenseNet121-RNN	COVID-19	99.99	99.78	99.57	100.0	99.78
	Pneumonia		99.78	100.0	99.57	99.78
	Normal		99.78	99.78	99.78	99.78
InceptionV3-RNN	COVID-19	99.95	98.56	99.35	96.44	97.87
	Pneumonia		98.56	96.32	99.55	97.91
	Normal		98.56	100.0	99.78	99.89
Inception-ResNetV2- RNN	COVID-19	99.99	99.50	98.49	100.0	99.24
	Pneumonia		99.50	100.0	99.72	99.86
	Normal		99.50	100.0	99.78	99.89

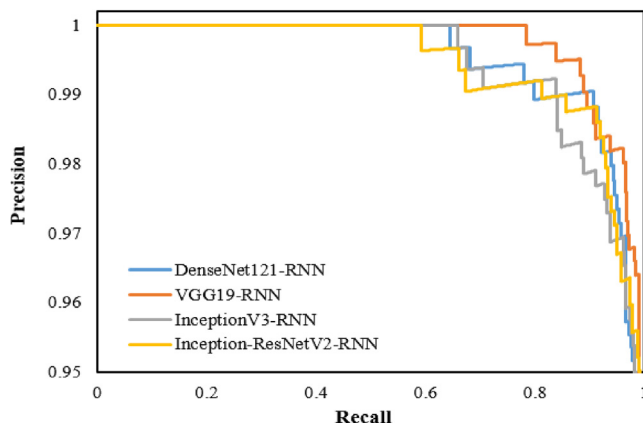**Fig. 7.** ROC curve of four combined CNN-RNN networks.

Fig. 8. PR curve of four combined CNN-RNN networks.

Table 6
Comparison between pre-trained CNN with CNN-RNN architecture based on COVID-19 patients.

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
VGG19	99.63	99.51	99.65	99.58
DenseNet121	99.26	99.45	99.38	99.41
InceptionV3	99.13	98.71	98.92	98.81
Inception-ResNetV2	99.28	99.38	98.06	98.72
VGG19-RNN	99.86	99.78	99.78	99.78

systems. In addition, a comparison in terms of computational time showed that [24] took 2641.0s for training 40 images and 4.0s for testing 10 images, [60] consumed 2277.6s for training 8997 images, [61] required 79184.3s and 262.0s for training and testing 4449 and 1638 images respectively. In our experiment, VGG19-RNN architecture took 16722.4s and 129.7s for training and testing 5551 and 1388 images respectively which is comparatively faster than other existing models. Hence, finally, it is evident that the VGG19-RNN network showed good performance compared to other studies.

5. Conclusion

During the COVID-19 pandemic, the use of deep learning techniques for the diagnosis of COVID-19 has become a crucial issue to overcome the limitation of medical resources. In this work, we used CNN with deep transfer learning and RNN to classify the X-ray samples into three categories: pneumonia, COVID-19, and normal. The four popular CNN networks were used to extract features, which were then applied by the RNN network to identify different classes. The VGG19-RNN is considered the best network with 99.9% accuracy, 99.9% AUC, 99.8% recall, and 99.8% F1-score to detect COVID-19 cases. Hopefully, it would reduce the workload for the doctor to test COVID-19 cases.

There are some limitations to our proposed system. First, the COVID-19 samples are small that need to be updated with more samples to validate our proposed system. Second, this experiment only works with a posterior–anterior view of chest X-ray, hence it is not able to effectively classify other views such as apical, lordotic, etc. Third, the performance of our experiment is not compared with radiologists which would be our future work.

Funding statement

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.

pneumonia and normal cases and is considered as a main deep learning architecture. A comparison between simple CNN-based pre-trained networks with our study is demonstrated in Table 6. It is clearly shown that the VGG19-RNN network has obtained higher performance than pre-trained CNN networks. Finally, another comparison between recent works with our study is demonstrated in Table 7. It is observed that existing systems can distinguish coronavirus infection with accuracy in the range of 80.6% to 99.6%. On the contrary, the VGG19-RNN network obtained 99.9% accuracy which is higher than other existing
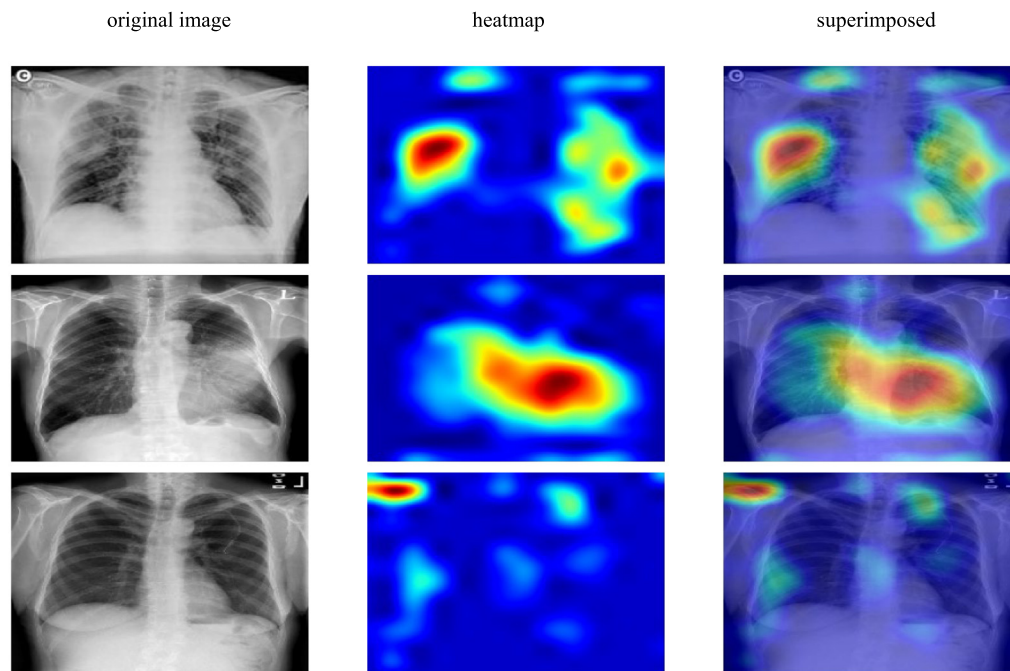
original image   heatmap   superimposed

Fig. 9. First, second and third rows represent the samples of COVID-19, pneumonia, and normal correspondingly. Besides, the first, second, and third columns refer to the original, heatmap and superimposed images for VGG19-RNN.

Table 7
Comparative study of the proposed CNN-RNN architecture with existing works concerning accuracy.

Author	Architecture	Accuracy (%)	COVID-19 accuracy (%)	Training (s)	Testing (s)
Luz et al. [17]	EfficientNet	93.9	–	–	–
Rahimzadeh and Attar [18]	Xception - ResNet50V2	91.4	99.6	–	–
Punn and Agarwal [20]	NASNetLarge	97.0	–	–	–
Khan et al. [21]	CoroNet (Xception)	89.5	96.6	–	–
Wang and Wong [22]	Tailored CNN	92.3	80.0	–	–
Narin et al. [23]	ResNet50	98.6	–	–	–
Hemdan et al. [24]	VGG19	90.0	–	2641.0	4.0
Apostolopoulos and Mpesiana [25]	VGG19	93.5	–	–	–
Loey et al. [27]	GoogleNet	80.6	100.0	–	–
Kumar and Kumari [28]	ResNet50-SVM	95.4	–	–	–
Bukhari et al. [29]	ResNet50	98.2	–	–	–
Abbas et al. [30]	DeTrac	95.1	–	–	–
Islam et al. [31]	CNN-LSTM	97.0	–	–	–
Faisal et al. [32]	VGG-19	99.5	100.0	–	–
Dey et al. [33]	Kapur's Entropy	90.0	–	–	–
Singh et al. [35]	SVM	99.6	–	–	–
Ucar and Korkmaz [60]	COVIDiagnosis-Net	98.3	100.0	2277.6	–
Asnaoui et al. [61]	Inception-ResNetV2	92.2	–	79184.3	262.0
Li et al. [62]	DenseNet	88.9	79.2	–	–
Chowdhury et al. [63]	Sgdm-SqueezeNet	98.3	96.7	–	–
Yang et al. [64]	VGG16	99.0	–	–	–
Suppakitjanusant et al. [65]	VGG19	85.0	–	–	–
Zhao et al. [66]	Bit-M	99.2	–	–	–
Sharmila et al. [67]	DCGANs	98.6	–	–	–
Reis et al. [68]	COVID-DSNet	97.6	–	–	–
Das et al. [69]	Ensemble method	91.62	–	–	–
Proposed system	**VGG19-RNN**	**99.9**	**99.9**	**16722.4**	**129.7**

References

[1] About Worldometer COVID-19 data - Worldometer. https://www.worldometers.info/coronavirus/ (Accessed 01 2023).

[2] Advice for the public. https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public (Accessed 01 2023).

[3] Everything about the Corona virus - Medicine and Health. (Accessed 01 2023).

[4] T. Ai, Z. Yang, L. Xia, Correlation of chest CT and RT-PCR testing in Coronavirus disease, Radiology 2019 (2020) 1–8, http://dx.doi.org/10.14358/PERS.80.2.000.

[5] C. Long, H. Xu, Q. Shen, X. Zhang, B. Fan, C. Wang, B. Zeng, Z. Li, X. Li, H. Li, Diagnosis of the Coronavirus disease (COVID-19): rRT-PCR or CT? Eur. J. Radiol. 126 (2020) 108961, http://dx.doi.org/10.1016/j.ejrad.2020.108961.

[6] H. Shi, X. Han, N. Jiang, Y. Cao, O. Alwalid, J. Gu, Y. Fan, C. Zheng, Articles Radiological findings from 81 patients with COVID-19 pneumonia in Wuhan, China: a descriptive study, Lancet Infect. Dis. 20 (2020) 425–434, http://dx.doi.org/10.1016/S1473-3099(20)30086-4.

[7] Z.Y. Zu, M. Di Jiang, P.P. Xu, W. Chen, Q.Q. Ni, G.M. Lu, L.J. Zhang, Coronavirus disease 2019 (COVID-19): A perspective from China, Radiology (2020) 200490, http://dx.doi.org/10.1148/radiol.2020200490.

[8] G.D. Rubin, C.J. Ryerson, L.B. Haramati, N. Sverzellati, J.P. Kanne, S. Raoof, N.W. Schluger, A. Volpi, J.-J. Yim, I.B.K. Martin, D.J. Anderson, C. Kong, T. Altes, A. Bush, S.R. Desai, J. Goldin, J.M. Goo, M. Humbert, Y. Inoue, H.-U. Kauczor, F. Luo, P.J. Mazzone, M. Prokop, M. Remy-Jardin, L. Richeldi, C.M. Schaefer-Prokop, N. Tomiyama, A.U. Wells, A.N. Leung, The role of chest imaging in patient management during the COVID-19 pandemic, Chest (2020) 1–11, http://dx.doi.org/10.1016/j.chest.2020.04.003.

[9] F. Shaheen, B. Verma, M. Asafuddoula, Impact of Automatic Feature Extraction in Deep Learning Architecture, 2016 Int. Conf. Digit. Image Comput. Tech. Appl.

DICTA 2016, 2016.

[10] A. Asraf, M.Z. Islam, M.R. Haque, M.M. Islam, Deep learning applications to combat novel Coronavirus (COVID-19) pandemic, SN Comput. Sci. 1 (2020) 1–7, http://dx.doi.org/10.1007/s42979-020-00383-w.

[11] P. Saha, M.S. Sadi, M.M. Islam, EMCNet: Automated COVID-19 diagnosis from X-ray images using convolutional neural network and ensemble of machine learning classifiers, Informatics Med. Unlocked. 22 (2021) 100505, http://dx.doi.org/10.1016/j.imu.2020.100505.

[12] Y. Guo, Y. Liu, E.M. Bakker, Y. Guo, M.S. Lew, CNN-RNN: a large-scale hierarchical image classification framework, Multimedia Tools Appl. 77 (2018) 10251–10271, http://dx.doi.org/10.1007/s11042-017-5443-x.

[13] Q. Yin, R. Zhang, X. Shao, CNN and RNN mixed model for image classification, MATEC Web Conf. 277 (2019) 02001, http://dx.doi.org/10.1051/matecconf/201927702001.

[14] T. Nakamura, T. Higuchi, H. Sawada, Phosphorylation of 6 mercaptopurine in leukemic cells, J. Japan Soc. Cancer Ther. 13 Th Cong (1976) 245–246.

[15] L.J. Muhammad, M.M. Islam, S.S. Usman, S.I. Ayon, Predictive data mining models for novel Coronavirus (COVID-19) infected patients' recovery, SN Comput. Sci. 1 (2020) 206, http://dx.doi.org/10.1007/s42979-020-00216-w.

[16] T. Mahmud, M.A. Rahman, S.A. Fattah, CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization, Comput. Biol. Med. 122 (2020) 103869, http://dx.doi.org/10.1016/j.compbiomed.2020.103869.

[17] E. Luz, P.L. Silva, R. Silva, L. Silva, G. Moreira, D. Menotti, Towards an effective and efficient deep learning model for COVID-19 patterns detection in X-ray images, 2020, pp. 1–10.

[18] M. Rahimzadeh, A. Attar, A New Modified Deep Convolutional Neural Network for Detecting COVID-19 from X-Ray Images, 2020, http://arxiv.org/abs/2004.08052.

[19] S. Minaee, R. Kafieh, M. Sonka, S. Yazdani, G.J. Soufi, Deep-COVID: Predicting COVID-19 from Chest X-Ray Images using Deep Transfer Learning, 2020, http://arxiv.org/abs/2004.09363.

[20] N.S. Punn, S. Agarwal, Automated diagnosis of COVID-19 with limited posteroanterior chest X-ray images using fine-tuned deep neural networks, 2020, http://arxiv.org/abs/2004.11676.

[21] A.I. Khan, J.L. Shah, M. Bhat, CoroNet: A Deep Neural Network for Detection and Diagnosis of Covid-19 from Chest X-ray Images, 2020, http://arxiv.org/abs/2004.04931.

[22] L. Wang, A. Wong, COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images, 2020, http://arxiv.org/abs/2003.09871.

[23] A. Narin, C. Kaya, Z. Pamuk, Zonguldak Bulent Ecevit University, 67100, Zonguldak, Turkey Department of Biomedical Engineering, 2020, ArXiv Prepr., arXiv:2003.10849 https://arxiv.org/abs/2003.10849.

[24] E.E.-D. Hemdan, M.A. Shouman, M.E. Karar, COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-ray Images, 2020, http://arxiv.org/abs/2003.11055.

[25] I.D. Apostolopoulos, T.A. Mpesiana, Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks, Phys. Eng. Sci. Med. (2020) 1–8, http://dx.doi.org/10.1007/s13246-020-00865-4.

[26] M.J. Horry, S. Chakraborty, M. Paul, A. Ulhaq, B. Pradhan, X-ray image based COVID-19 detection using pre-trained deep learning models, 2007.

[27] M. Loey, F. Smarandache, N.E.M. Khalifa, Within the lack of chest COVID-19 X-ray dataset: A novel detection model based on GAN and deep transfer learning, Symmetry (Basel) 12 (2020) http://dx.doi.org/10.3390/SYM12040651.

[28] P. Kumar, S. Kumari, Detection of coronavirus disease (COVID-19) based on deep features, 9, 2020, http://dx.doi.org/10.20944/preprints202003.0300.v1, https://www.preprints.org/manuscript/202003.0300/V1.

[29] S.U.K. Bukhari, S.S.K. Bukhari, A. Syed, S.S.H. Shah, The diagnostic evaluation of Convolutional Neural Network (CNN) for the assessment of chest X-ray of patients infected with COVID-19, 2020, http://dx.doi.org/10.1101/2020.03.26.20044610, MedRxiv. 2020.03.26.20044610.

[30] A. Abbas, M.M. Abdelsamea, M.M. Gaber, Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network, 2020, http://arxiv.org/abs/2003.13815.

[31] M.Z. Islam, M.M. Islam, A. Asraf, A Combined Deep CNN-LSTM Network for the Detection of Novel Coronavirus (COVID-19) Us- ing X-ray Images, 2020, pp. 1–20, http://dx.doi.org/10.1101/2020.06.18.20134718.

[32] N.B. Prakash, N. Murugappan, G.R. Hemalakshmi, M. Jayalakshmi, M. Mahmud, Deep transfer learning for COVID-19 detection and infection localization with superpixel based segmentation, Sustain. Cities Soc. 75 (2021) 103252, http://dx.doi.org/10.1016/j.scs.2021.103252.

[33] N. Dey, V. Rajinikanth, S.J. Fong, M.S. Kaiser, M. Mahmud, Social group optimization–Assisted Kapur's entropy and morphological segmentation for automated detection of COVID-19 infection from computed tomography images, Cognit. Comput. 12 (2020) 1011–1023, http://dx.doi.org/10.1007/s12559-020-09751-3.

[34] V.N.M. Aradhya, M. Mahmud, D.S. Guru, B. Agarwal, M.S. Kaiser, One-shot cluster-based approach for the detection of COVID–19 from chest X–ray images, Cognit. Comput. 13 (2021) 873–881, http://dx.doi.org/10.1007/s12559-020-09774-w.

[35] A.K. Singh, A. Kumar, M. Mahmud, M.S. Kaiser, A. Kishore, COVID-19 infection detection from chest X-ray images using hybrid social group optimization and support vector classifier, Cognit. Comput. (2021) http://dx.doi.org/10.1007/s12559-021-09848-3.

[36] M. Shamim Kaiser, M. Mahmud, M.B.T. Noor, N.Z. Zenia, S. Al Mamun, K.M. Abir Mahmud, S. Azad, V.N. Manjunath Aradhya, P. Stephan, T. Stephan, R. Kannan, M. Hanif, T. Sharmeen, T. Chen, A. Hussain, IWorksafe: Towards healthy workplaces during COVID-19 with an intelligent phealth app for industrial settings, IEEE Access 9 (2021) 13814–13828, http://dx.doi.org/10.1109/ACCESS.2021.3050193.

[37] J.P. Cohen, P. Morrison, L. Dao, Covid-19 image data collection, 2023, (Accessed 01 2023).

[38] COVID-19 chest X-ray. https://github.com/agchung (Accessed 01 2023).

[39] Radiopaedia. COVID-19 X-ray Cases. 2023 (Accessed 01 2023).

[40] COVID-19 DATABASE / SIRM. https://www.sirm.org/en/category/articles/covid-19-database/ (Accessed 01 2023).

[41] COVID-19 Chest X-ray Image Repository. https://figshare.com/articles/COVID-19_Chest_X-ray_Image_Repository/12580328/2 (Accessed 01 2023).

[42] COVID-19 Image Repository. https://figshare.com/articles/COVID-19_Image_Repository/12275009/1 (Accessed 01 2023).

[43] Mendeley Data - Augmented COVID-19 X-ray Images Dataset. https://data.mendeley.com/datasets/2fxz4px6d8/4 (Accessed 01 2023).

[44] Chest X-ray Images (Pneumonia)/Kaggle, Kaggle. https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia (Accessed 01 2023).

[45] NIH Chest X-rays/Kaggle. https://www.kaggle.com/nih-chest-xrays/data? (Accessed 01 2023).

[46] M. Huh, P. Agrawal, A.A. Efros, What makes ImageNet good for transfer learning?, 2016.

[47] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., 2015, pp. 1–14.

[48] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM. 60 (2017) 84–90, http://dx.doi.org/10.1145/3065386.

[49] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017. 2017-(2017), 2017, pp. 2261–2269, http://dx.doi.org/10.1109/CVPR.2017.243.

[50] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2016-, 2016, pp. 2818–2826, http://dx.doi.org/10.1109/CVPR.2016.308.

[51] U. Nazir, N. Khurshid, M.A. Bhimra, M. Taj, Tiny-Inception-ResNet-v2: Using Deep Learning for Eliminating Bonded Labors of Brick Kilns in South Asia, 2019, http://arxiv.org/abs/1907.05552.

[52] P.J. Werbos, Backpropagation through time: What it does and how to do it, Proc. IEEE. 78 (1990) 1550–1560, http://dx.doi.org/10.1109/5.58337.

[53] Yoshua. Bengio, Patrice. Simard, Paolo. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE Trans. Neural Netw. 5 (2014) 157.

[54] S. Hochreiter, The vanishing gradient problem during learning recurrent neural nets and problem solutions, Int. J. Uncertainty, Fuzziness Knowledge-Based Syst. 6 (1998) 107–116, http://dx.doi.org/10.1142/S0218488598000094.

[55] P. Liu, X. Qiu, X. Chen, S. Wu, X. Huang, Multi-timescale long short-term memory neural network for modelling sentences and documents, in: Conf. Proc. - EMNLP 2015 Conf. Empir. Methods Nat. Lang. Process, 2015, pp. 2326–2335, http://dx.doi.org/10.18653/v1/D15-1280.

[56] M.J. Brown, L.A. Hutchinson, M.J. Rainbow, K.J. Deluzio, A.R. De Asha, A comparison of self-selected walking speeds and walking speed variability when data are collected during repeated discrete trials and during continuous walking, J. Appl. Biomech. 33 (2017) 384–387, http://dx.doi.org/10.1123/jab.2016-0355.

[57] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, 2012, pp. 1–18.

[58] D.M. Hawkins, The problem of overfitting, J. Chem. Inf. Comput. Sci. 44 (2004) 1–12, http://dx.doi.org/10.1021/ci0342472.

[59] D.P. Kingma, J.L. Ba, Adam: A method for stochastic optimization, in: 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., 2015, pp. 1–15.

[60] F. Ucar, D. Korkmaz, COVIDiagnosis-Net: Deep Bayes-SqueezeNet based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images, Med. Hypotheses. 140 (2020) http://dx.doi.org/10.1016/j.mehy.2020.109761.

[61] K. El Asnaoui, Y. Chawki, Using X-ray images and deep learning for automated detection of coronavirus disease, J. Biomol. Struct. Dyn. (2020) 1–12, http://dx.doi.org/10.1080/07391102.2020.1767212.

[62] X. Li, C. Li, D. Zhu, COVID-MobileXpert: On-Device COVID-19 Screening using Snapshots of Chest X-ray, 2020, http://arxiv.org/abs/2004.03042.

[63] M.E.H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M.A. Kadir, Z. Bin Mahbub, K.R. Islam, M.S. Khan, A. Iqbal, N. Al-Emadi, M.B.I. Reaz, Can AI help in screening viral and COVID-19 pneumonia?, 2020, http://arxiv.org/abs/2003.13145.

[64] D. Yang, C. Martinez, L. Visuña, H. Khandhar, C. Bhatt, J. Carretero, Detection and analysis of COVID-19 in medical images using deep learning techniques, Sci. Rep. 11 (2021) 1–13, http://dx.doi.org/10.1038/s41598-021-99015-3.

[65] P. Suppakitjanusant, S. Sungkanuparph, T. Wongsinin, S. Virapongsiri, N. Kasemkosin, L. Chailurkit, B. Ongphiphadhanakul, Identifying individuals with recent COVID-19 through voice classification using deep learning, Sci. Rep. 11 (2021) 1–7, http://dx.doi.org/10.1038/s41598-021-98742-x.

[66] W. Zhao, W. Jiang, X. Qiu, Deep learning for COVID-19 detection based on CT images, Sci. Rep. 11 (2021) 1–12, http://dx.doi.org/10.1038/s41598-021-93832-2.

[67] S.V.J., J.F.D., Deep learning algorithm for COVID-19 classification using Chest X-ray images, Comput. Math. Methods Med. 2021 (2021) 9269173, http://dx.doi.org/10.1155/2021/9269173.

[68] H.C. Reis, V. Turk, COVID-DSNet: A novel deep convolutional neural network for detection of coronavirus (SARS-CoV-2) cases from CT and Chest X-ray images, Artif. Intell. Med. 134 (2022) 102427, http://dx.doi.org/10.1016/j.artmed.2022.102427.

[69] 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, http://dx.doi.org/10.1007/s10044-021-00970-4.