A CNN-LSTM network with multi-level feature extraction-based approach for automated detection of coronavirus from CT scan and X-ray images

Hamad Naeem *, Ali Abdulqader Bin-Salem

School of Computer Science and Technology, Zhoukou Normal University, Zhoukou 466001, Henan, China

ARTICLE INFO

ABSTRACT

Auto-detection of diseases has become a prime issue in medical sciences as population density is fast growing. An intelligent framework for disease detection helps physicians identify illnesses, give reliable and consistent results, and reduce death rates. Coronavirus (Covid-19) has recently been one of the most severe and acute diseases in the world. An automatic detection framework should therefore be introduced as the fastest diagnostic alternative to avoid Covid-19 spread. In this paper, an automatic Covid-19 identification in the CT scan and chest X-ray is obtained with the help of a combined deep learning and multi-level feature extraction methodology. In this method, the multi-level feature extraction approach comprises GIST, Scale Invariant Feature Transform (SIFT), and Convolutional Neural Network (CNN) extract features from CT scans and chest X-rays. The objective of multi-level feature extraction is to reduce the training complexity of CNN network, which significantly assists in accurate and robust Covid-19 identification. Finally, Long Short-Term Memory (LSTM) along the CNN network is used to detect the extracted Covid-19 features. The Kaggle SARS-CoV-2 CT scan dataset and the Italian SIRM Covid-19 CT scan and chest X-ray dataset were employed for testing purposes. Experimental outcomes show that proposed approach obtained 98.94% accuracy with the SARS-CoV-2 CT scan dataset and 83.03% accuracy with the SIRM Covid-19 CT scan and chest X-ray dataset. The proposed approach helps radiologists and practitioners to detect and treat Covid-19 cases effectively over the pandemic.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

This outbreak of the coronavirus is placing all industries on lockdown. According to the current data of the World Health Organization (WHO), more than 12 million people have been infected, with roughly 552,050 deaths as of 9 July 2020 [1]. Also, health services have hit a point of decline in advanced nations, causing intensive care units to lack. Two different coronaviruses, severe acute respiratory syndrome (SARS) and the middle east respiratory syndrome (MERS), were identified for the strain which started to sprout in Wuhan, China [2]. The Covid-19 signs may include cold to fever, short breathing, and acute respiratory disease [3]. Compared with SARS, both coronavirus and the respiratory system damage the kidneys and liver [4]. For preventing Covid-19 and deadly diseases, no definite vaccine is available. The only way to avoid transmitting the virus infection among safe individuals is to separate the infected individual. RT-PCR is a method that can identify Covid-19 from a respiratory sample [5]. Because of the restricted supply and limitations of RT-PCR, the

avoidance of coronaviral infection has been difficult. Radiological imaging methods in comparison are used to diagnose SARS-CoV-2 by integrating clinical indications of infected individuals with laboratory reports [6]. Radiological imagery, including lung radiation and chest CT, is helpful in promptly eliminating infectious individuals and managing the epidemic [7]. Such techniques can quickly identify the coronavirus radiological features.

The chest X-rays are the first choice of radiologists since most hospitals have X-rays facilities [8]. However, it may be challenging to differentiate between soft tissues from X-ray images [9]. Chest CT scan is widely used to solve this problem. Radiologists are needed to interpret the CT scans of the chest [10]. However, it requires time, and the task is vulnerable to error. Automatic Covid-19 detection from both CT scan and chest X-ray is thus needed. Machine learning has also seen a boom in its utilization in a wide variety of medical areas, including detection of coronary disorders [11,12], diabetic retinopathy classification [13], and cornel pattern classification [14]. These technologies demonstrate the potential to reduce medical errors, find or track asymptomatic carriers early, and provide patients with appropriate health services to improve their treatment. Computer vision [15], machine learning [16–18], and deep learning [19,20] have recently been

employed for automatically detecting different human diseases, thus ensuring intelligent healthcare [21,22]. The deep learning approach acts as a feature extractor that improves the precision of classification [23]. Examples of the contributions [24,25] from deep learning are identification of the lung tumor, bone suppression by radiography, diabetic retinopathy, prostate segmentation, skin lesions, and myocardium involvement in coronary CT scans.

1.1. Challenges in Automated Covid-19 Detection

Our objective is to detect and classify Covid-19 in CT scan and X-ray images. To do this, the main challenges that must be addressed in artificial intelligence and Covid-19 detection are as follows.

- The majority of recent solutions focus on Covid-19 identification, which can only facilitate doctors in detecting Covid-19 patients among other Pneumonia patients. They are unable to determine the severity in Covid-19 cases. Therefore, radiologists and physicians need more time to finish this task.
- Collecting a large labeled dataset is the most challenging task in automated Covid-19 detection. The number of non-severe cases is much larger than that of severe cases as per the recent Covid-19 literature. Overfitting from imbalanced classes in the Covid-19 dataset significantly reduces the accuracy of automatic detection [26].
- CNN networks can extract features from large datasets of high-dimensional images and videos. However, the training process of these networks is more complex, time-consuming, and resource-consuming due to high dimensional input. As a result, the need for these networks finishes, especially in IoT-based medical solutions where the computing resources are minimal compared to CNN computational complexity [27,28].

1.2. Contributions

In this paper, a combined deep learning and multi-level features extraction-based approach is designed to detect and classify Covid-19 CT scans and chest X-rays. In this article, the significant contributions are:

- We propose a combined deep learning and multi-level features extraction-based approach to automatically identify the soft tissues in CT scan and chest X-ray images of Covid-19. Our approach supports radiologists and doctors in detecting Covid-19 automatically and categorizing severity.
- The multi-level features extraction-based strategy comprises GIST, Scale Invariant Feature Transform (SIFT), and Convolutional Neural Network (CNN) is proposed to reduce CNN network training complexity. It helps to minimize parameter sizes and input dimensions for the CNN network. A combined CNN-LSTM based deep learning strategy is proposed for automatic Covid-19 detection with severity classification. The comparison with other state-of-the-art deep learning methodologies confirms the feasibility of the proposed approach.
- The fused feature set is visualized using t-distributed Stochastic Neighbor Embedding (t-SNE) to confirm that the multi-level features extracted from the proposed strategy are not sparse. The visualization results show that the fused feature set is denser and can effectively perform deep learning tasks.

- The overfitting and imbalanced data problems significantly affect the accuracy and loss values of the deep learning model. Dropout layer and activation function are used in the proposed strategy for solving the overfitting challenge, whereas the imbalanced data problem is solved by using the Synthetic Minority Oversampling (SMOTE).

The remaining sections of the paper are organized as follows: the related works are given in Section 2, the proposed methodology is explained in Section 3, results and discussions are provided in Section 4, and finally, the conclusion is given in Section 5.

2. Related works

More studies have been conducted on deep learning-based methods to achieve high Covid-19 detection and severity classification performance. For example, Sarker et al. [29] used DenseNet−121 to apply transfer learning for Covid-19 identification. They designed a platform that analyzes radiological images and displays the areas that are affected. This method had obtained 87% accuracy. Shan et al. [30] developed a deep learning method to classify infected lung regions automatically. The methodology was evaluated on 300 coronavirus-infected cases. The accuracy of this methodology was 91%. The proposed approach could not determine any further severity of the Pneumonia. Zhang et al. [31] utilized DenseNet for diagnosing coronaviral disease. Case identification sensitivities of Covid-19 and non-Covid-19 were 96% and 70.65%, respectively. Wang et al. [32] employed pre-trained deep learning methods to recognize the lung images of Covid-19. This method was tested on 1266 patients in six cities. This method had an accuracy of 87%.

Kumar et al. [33] proposed a deep learning approach for detecting Covid-19 chest X-ray images. Nine pre-trained deep learning models were employed to extract features from Covid-19 X-rays, and SVM was used to classify them. This method had an accuracy of 95.38%. Aayush et al. [34] used chest CT scan images and executed a pre-trained deep neural network architecture to diagnose Covid-19 among patients. They further used the DenseNet201 model for deep transfer learning (DTL) to determine whether or not a patient was infected with Covid-19. To extract features from Covid-19 CT scans, their CNN network used pre-determined learning weights on the ImageNet dataset. They trained their deep learning network with direct image input and 14 layers. Experimental results of their predictive model outperformed state-of-the-art approaches. Silva et al. [35] used a voting-based deep learning strategy to identify Covid-19 positive patients. They utilized transfer learning along with a voting strategy. They trained their deep learning network with direct image input and 18 layers. Their voting-based system categorized Covid-19 images into different groups.

Furthermore, they tested their strategy on large Covid-19 CT scan image datasets. Their approach achieved 86.79% accuracy on SARS-CoV-2 CT scan dataset. Alshazly et al. [36] utilized the advanced deep learning models for transfer learning on Covid-19 CT scan images. They performed extensive experiments on two large CT scan image datasets consisting of covid positive and negative patients. They trained their deep learning networks with direct image input and 201 layers. Experimental evaluation showed that their model achieved an average accuracy of 99.4% on the SARS-CoV-2 dataset and 92.9% on the Covid-19 CT scan dataset. Angelov et al. [37] built a public SARS-CoV-2 dataset based on CT scan images of suspected patients. They further proposed an eXplainable deep learning strategy (xDNN) to identify positive and negative covid patients. They had used direct input images for their pre-trained network. The baseline experiment of their public dataset achieved an excellent accuracy of 97.38% with minimal accuracy loss.

Covid-19 severity detection is one of the most challenging tasks in assessing the condition of a covid positive patient. Deep learning-based severity assessment has been considered in several studies to be more effective and quantitative than report-based radiologist assessment. Despite the promising results of deep learning methods on Covid-19 severity diagnosis, few studies on computer vision-based covid detection have been published. For example, Yue et al. [38] used 729 CT scan images of Covid-19 patients to measure the severity. They used a pre-trained deep neural network to classify the severity and achieved an overall detection accuracy of 95.34%. They had used direct input images for their pre-trained network. Carvalho et al. [39] categorized covid patients into mild, moderate, and severe groups to perform severity detection. The experimental evaluation of artificial neural network (ANN) achieved an overall detection accuracy of 82% on 300 CT scan images of covid positive patients. They had used direct input images for network training. Xiao et al. [40] used a residual CNN (ResNet34) model to optimize their deep neural network for Covid-19 disease severity progression. They had used direct input images for their pre-trained network. The empirical analysis of 408 Covid-19 positive patients yielded an overall detection accuracy of 81.9%. Our proposed work fills the following research gaps.

- CNN is usually quite promising to extract features from high dimensional input such as images etc. Even though the deep learning methodologies described in published works have proven excellent [29–33,39], the training period for such approaches is exceptionally long, as their CNN networks require large-scale optimization processes. It is due to the Covid-19 image's large input size. Fine-tuning or transfer learning is used to tackle this problem in published researches [34–38,40]. The ImageNet dataset was used in the pre-training of most of these methods. Therefore, the training complexity for the CNN network will rise during fine-tuning if training and testing scenes are part of entirely different applications. This limitation poses a substantial challenge for IoT-based medical applications with limited processing resources than CNN computational complexity. Our paper presents a new way to decrease CNN's training complexity by reducing parameter sizes and input dimensions. The results reported in Tables 4 and 5, and Fig. 9 correspondingly justify this claim.
- The majority of published research focuses on detecting Covid-19, which can only aid physicians in identifying Covid-19 patients among other patients. Covid-19 severity assessment can also assist physicians in prioritizing patient treatment. Usually, a lengthy evaluation time is required for radiologists and physicians to fulfill both tasks. However, deep learning-based solutions can provide fast, efficient, and accurate identification of Covid-19 with severity assessment. A few studies in the literature have focused on the severity assessment of Covid-19, and they have attained less accurate severity assessment results due to the insufficient number of Covid-19 samples. As a result, the automatic Covid-19 severity assessment is still under-examined, requiring additional research in this area. However, our research focuses on both the detection and the severity classification of Covid-19. The results can be viewed in Table 2.

3. Proposed methodology

3.1. Outline of proposed methodology

The proposed approach is designed to detect and classify Covid-19 in CT scans and chest X-rays. We collected a SARS-CoV-2 CT scan dataset (www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset) from the Kaggle database for experimental purposes. The dataset contains a total of 2390 CT scans, 1229 of which are positive for infection with SARS-CoV-2, and the remaining 1161 are negative for infection with SARS-CoV-2. 34 of the Covid-19 cases were male, while 28 of them were female. The Italian Society of Medical and Interventional Radiology (SIRM) Covid-19 database provided the second dataset for this study (https://www.sirm.org/category/senza-categoria/Covid-19/). It includes 220 CT scan and X-ray images of the lungs from 79 actual Covid-19 cases. We analyzed 79 cases in 115 real cases of Covid-19. Senior radiologists and specialists of SIRM with several years of expertise in analyzing CT scans and chest X-rays have provided brief history of each Covid-19 case in the database. The second dataset was used in this investigation to diagnose the severity of Covid-19. Fig. 1 shows a chunk of lung CT scans and chest X-rays of Covid-19 cases. According to the National Health Commission of China's directions on Covid-19, the opacity and lung disease involvement produce legitimate results regarding Covid-19 disease severity [41]. The severity level of Covid-19 was evaluated using the lung severity score of Warren et al. [42]. Each lung sample was grouped according to its particular lung involvement score (0–4). Based on the illness stage and lung severity score, we divided the 79 real Covid-19 cases into four groups (Mild, Moderate, Severe, and Critical). Score 0 represents no lung involvement and opacity.

- Mild Case (Score 1) = Pneumonia affects 25% of the lung region (Opacity is present in the lower and middle lung sections). The period of illness ranges from 1 to 10 days.
- Moderate Case (Score 2) = Pneumonia affects 25%–50% of the lung region (In the lower and middle lung sections, opacity is present). The period of illness ranges from 11 to 20 days.
- Severe Case (Score 3) = Pneumonia affects 50%–75% of the lung region (Opacity exists in all lung sections). The period of illness ranges from 21 to 30 days.
- Critical Case (Score 4) = Pneumonia affects 75%–100% of the lung region (Opacity exists in all lung sections). The period of illness ranges more than 30 days.

The proposed approach for automatic Covid-19 detection and classification has five significant steps: (a) Level 1 — Feature Extraction, Selection, and Fusion (Handcrafted Features), (b) t-SNE Feature Visualization, (c) Data Augmentation, (d) Level 2 — Feature Extraction (CNN), (e) Automatic Covid-19 Detection and Classification (CNN-LSTM). Fig. 2 shows the flow diagram of the proposed approach. The proposed approach uses a combined CNN-LSTM strategy with multi-level features extraction-based strategy for Covid-19 detection and classification. The local and global features are extracted, selected, and fused from CT scans and chest X-rays using conventional image descriptors, i.e., SIFT and GIST in the level 1 feature extraction phase. Local and global features are further merged in a single feature vector with serial feature fusion. t-SNE is used to transform high dimensional fused feature set into low dimensional space. Synthetic Minority Over-sampling (SMOTE) [43] is used to handle the imbalanced data problem. Deep features are extracted from the fused feature set using the CNN network in the level 2 feature extraction phase. CNN extracts more valuable features with further reduced dimensions. Lastly, the combined CNN-LSTM model uses the derived features for Covid-19 detection and classification. Details on each stage are provided further below subsections.

3.2. Level 1 — feature extraction

CNN network is quite effective for a variety of high dimensional data such as images and videos etc. It needs a high-dimensional optimization method during which the training time

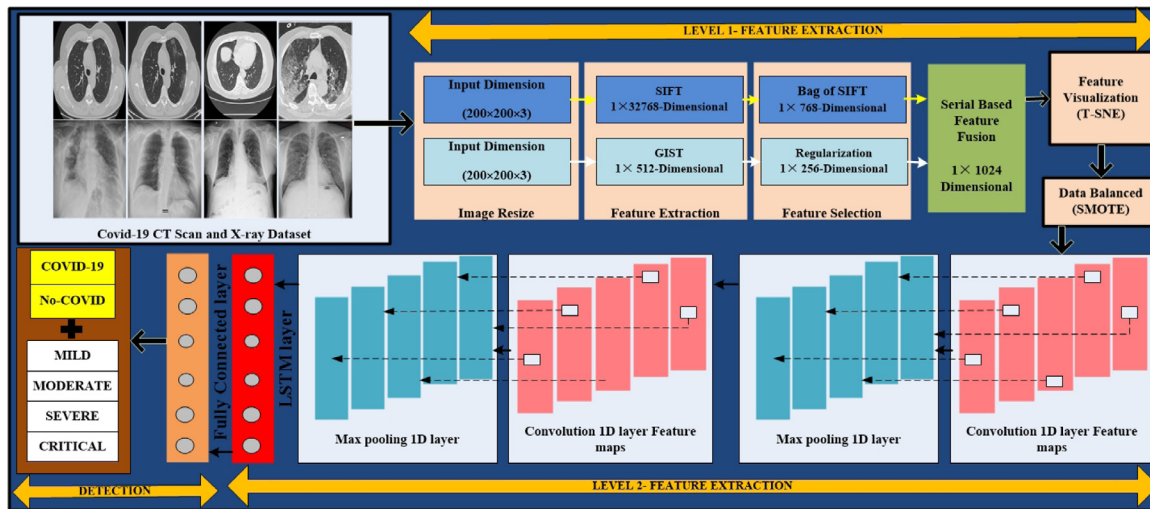**Fig. 1.** A chunk of lung CT scans and chest X-rays of Covid-19 cases.

Fig. 2. Workflow of proposed automatic Covid-19 detection and classification model.

is long since the input size is large. In addition, if testing samples are drastically different from training samples, then the CNN network must be retrained or fine-tuned for identification [27,28]. Level 1 feature extraction is used to reduce input dimensions of data for the CNN network. Consequently, it increases the CNN network training efficiency without loss of accuracy, minimizes input data dimensions and parameter sizes. Three significant steps are used to extract SIFT and GIST features of the CT scan and chest X-ray images, namely feature extraction, selection, and fusion. The description of each step is given below:

3.2.1. Extraction and selection of SIFT feature

The SIFT descriptor extracts local image features and uses an appropriate key point to describe them. Since CT scans and chest X-rays contain many textural, corners, and edges features, it is necessary to extract local features to minimize detection errors. Conventional SIFT descriptor [44] often fails to extract more local features from entire image patches. Due to certain key advantages, the proposed strategy chooses Dense SIFT Descriptor (D-SIFT) for local feature extraction from Covid-19 images. D-SIFT uses a dense grid for local features extraction from whole image patches. D-SIFT is x30-x60 times faster than conventional SIFT descriptor [45]. D-SIFT provides more high dimensional local characteristics of an image, and it is more accurate than

other local feature descriptors such as SURF [46] and LBP [47]. Several studies have previously preferred D-SIFT over other feature descriptors for extracting local features from medical image datasets [48].

The main workflow of the D-SIFT feature descriptor is presented below.

- In the first stage for feature description, D-SIFT selects the bounding box then uses a patch to move over CT scan and chest X-ray. The patch slides all areas within the bounding box to describe the feature. Each patch size is determined from $4 \times \eta_x \times 4 \times \eta_y$ where η_x and η_y depicts each cell size in the patch. The patch is known as a descriptor sampling region.
- The gradient histogram of the pixel is calculated in eight directions for each cell in the second stage. Each patch has 128-dimensional characteristics. All patch features are combined to deliver D-SIFT features.
- In the third stage, Eq. (1) defines the total dimensionality of D-SIFT features.

$$D = \text{ceil}\left(\frac{W - (4-1) \times \eta_x}{\delta_x}\right) \times \text{ceil}\left(\frac{W - (4-1) \times \eta_y}{\delta_y}\right) \times 128$$

(1)

where W depicts the width of the CT scan and chest X-ray, δ_y and δ_x depicts vertical and horizontal steps, η_y and η_x depicts the width and height of the cell.

Consequently, the overall D-SIFT dimensionality is $191 \times 191 \times 128$. The input dimensions of the Covid-19 image are $200 \times 200 \times 3$ in our case. The strategy is developed using MATLAB language.

The conventional Bag of Features (BOF) selection approach [49] is used to reduce the dimensionality of D-SIFT features and boost the efficiency of the training of CNN. The D-SIFT features are then grouped into clusters which are arbitrarily built to minimize overall dimensionality, and their frequency is known as the weight. The overall dimension of the D-SIFT features is reduced up to 768. The entire stages of the BOF feature selection approach are presented below.

- The most illustrative D-SIFT features are clustered into arbitrary clusters using the K-means approach [50] in the first level. K-means clustering initially selects the cluster centers arbitrarily and then uses the Euclidean distance to allocate the closest cluster centers to each D-SIFT feature. Each visual codeword refers to a cluster center. Eq. (2) is used to construct the final dictionary for CT scan and chest X-ray datasets.

$$\text{Dictionary} = \sum_{j=1}^{k} \sum_{i=1}^{n} \|x_i - c_j\|^2 \tag{2}$$

where k is the visual codeword (in our case, k = 768), n represents feature points'x_i number, and the centroids of cluster j represent c_j. Eq. (3) determines the functionality of the learning dictionary.

$$\text{Dictionary}_L = f_{kmeans}(D - Sift_{feature}, D_{size}) \tag{3}$$

where D_{size} represents the dictionary size.

- In the second level, each D-SIFT feature is quantized with the closest visual codeword in the dictionary. The value of histogram differs among [0,1] scales. The D-SIFT feature vector is computed after Euclidean distance calculation between local feature and visual dictionary, as shown in Eq. (4). 768-dimensional D-SIFT feature vector is obtained in the final level of feature selection.

$$D - Sift_{fv} = f_{histogram}(D - Sift_{feature}, Dictionary_L) \tag{4}$$

3.2.2. Extraction and selection of GIST feature

The extraction of image texture is referred to as global feature extraction. Many works suggest the GIST descriptor for extraction of global features inside the CT scan and chest X-ray due to its high classification performance. Therefore, for global feature extraction, we pick the GIST descriptor. GIST descriptor uses first Gabor multi-scale and multi-directional filtration and then extracts the image's 512-dimensional texture feature vector. Global feature extraction for CT scan and chest X-ray is performed using Eq. (5). The weighted combination of the output size of multi-scale-oriented filters signifies the value of the GIST feature vector that is computationally expensive. A regularization technique [51] is selected to reduce the dimensionality of the GIST feature vector. 256-dimensional GIST feature vector is obtained in the final level of feature selection.

$$\begin{aligned}
g_{mn}(x, y) &= a^{-m}g(x', y'), a > 1 \\
x' &= a^{-m}(x\cos\theta + y\cos\theta) \\
y' &= a^{-m}(-x\cos\theta + y\cos\theta) \\
\theta &= \frac{n\pi}{n+1}
\end{aligned} \tag{5}$$

where Gabor filter is indicated by g (x, y), the Scale factor of wavelet expansion is indicated by a^{-m}, the number of scales is

represented by m, the number of directions is represented by n, and filter direction is represented with θ.

D-SIFT and GIST descriptors concentrate on edges, corners, and textural feature points that define each feature's surrounding area. As a result, it is confirmed that both descriptors can preserve the most relevant information inside Covid-19 samples. Fig. 3 represents the most essential 10,50,100 and 300 D-SIFT and GIST features as red spots.

3.2.3. Feature fusion

D-SIFT and GIST feature descriptors offer two vectors with lengths of 1×768 and 1×256, respectively. The Eqs. (6) and (7) represent the features extracted by two feature descriptors, and Eq. (8) display a fused feature set of 1×1024 length. The proposed strategy chooses a serial-based feature fusion method for concatenating features of both descriptors. The resulting fused feature set provides a 1024-dimensional feature vector. The reduced input fused feature set significantly decreases the training complexity of the CNN network. The significance of fused feature set is further validated with t-SNE in Section 3.3.

$$\begin{aligned}
f_{d-sift_{1\times768}} = (d - sift_{1\times1}, d - sift_{1\times2}, d - sift_{1\times3}, \\
\ldots\ldots\ldots, d - sift_{1\times768})
\end{aligned} \tag{6}$$

$$f_{gist_{1\times256}} = \left(gist_{1\times1}, gist_{1\times2}, gist_{1\times3}, gist_{1\times4}, \ldots\ldots\ldots, gist_{1\times256} \right) \tag{7}$$

$$\text{Fused}_{fv_{1\times1024}} = \left\{ f_{d-sift_{1\times768}}, f_{gist_{1\times256}} \right\} \tag{8}$$

where $\text{Fused}_{fv_{1\times1024}}$ represents fused feature set.

3.3. t-SNE Feature visualization

The t-SNE feature visualization is used to determine whether the fused feature set contains essential or sparse information. Van der Maaten and Hinton introduced t-SNE [52] in 2008 as a popular approach of high-dimensional data visualization. The t-SNE algorithm transforms high dimensional data into low dimensional spaces. t-SNE algorithm has many tunable parameters such as the number of iterations and perplexity etc. The perplexity value of t-SNE is used to balance local and global data aspects in the resulting plot.

The balanced attention between local and global aspects of the fused feature vector for various perplexity values is seen in Fig. 4. We conducted two t-SNE visualization experiments using the R language. We approximate the lowest perplexity value required for separating Covid-19 (+) and Covid-19 (−) clusters in the first experiment. We visualized the finest Covid-19 (+) and Covid-19 (−) clusters separation through optimal perplexity value in the second experiment. t-SNE employs iterations to distinguish between different sample types. To visualize the separate Covid-19 (+) and Covid-19 (−) clusters, we used 300 iterations with each perplexity value. Fig. 4 displays the first row with dimensional reduced t-SNE plots of fused feature set on 1,2,3, and 4 perplexity values. Covid-19 (+) samples are indicated in an orange cluster, and Covid-19 (−) samples are indicated by a light blue-color cluster. The first row in Fig. 4 displays no visible cluster with overlapping samples when the perplexity value is 1. Even though on 2,3 and 4 perplexity values, the outcomes of the first t-SNE experiment shows separation of Covid-19 (+) and Covid-19 (−) samples into visual clusters but still the exception of a few outlier data points exists. Therefore, the second experiment is conducted for clear separation of Covid-19 (+) and Covid-19 (−) clusters. We chose four optimal perplexity values such as 5, 6, 7, and 8 to perform the second t-SNE experiment, as shown in the second row of Fig. 4. The second row in Fig. 4 shows the t-SNE plots with clearly distinct Covid-19 (+) and Covid-19 (−) clusters,
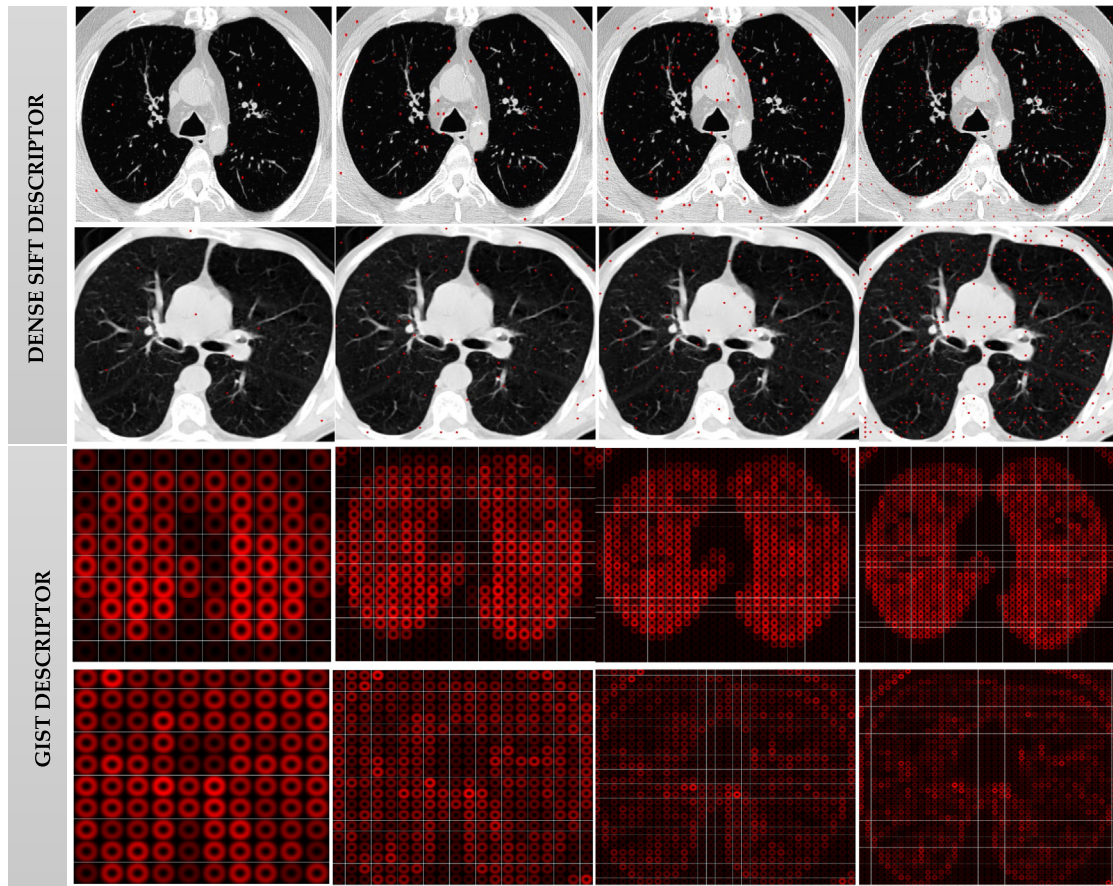
Fig. 3. Feature extraction using D-SIFT and GIST descriptor. Red dots represent the location of (from left to right:) 10, 50, 100, and 300 D-SIFT and GIST features (from top to bottom Covid and Non-Covid CT scans).

which display low dimensional fused feature set can categorize samples into their respective categories. The recommended high perplexity value ranges from 5 to 50. All the observations in Fig. 4 indicate that the fused feature set is dense with no missing values.

The density of the dataset dramatically affects the absolute accuracy of the prediction. Higher density generally leads to more accurate predictions because more expressive data is available for learning [53]. The t-SNE visual clusters are more isolated and improve classification performance. A dataset can be separated with appropriate perplexity value, which can also be categorized by suitable hyperparameters, as observed in the classification performance of the proposed CNN-LSTM model in Tables 2–5, and Table 6.

3.4. Data augmentation

In deep learning, we generally need to enter adequate data to train the model to avoid overfitting. Data augmentation can be used to enhance the sample size, reducing the

impact of unbalanced data. Overfitting issues can be efficiently avoided with the proper data augmentation strategy, improving the model's robustness. Several studies have recently been conducted to provide new data augmentation methods. For example, Jayadeva et al. [54,55] introduced a non-iterative method for adding samples to small datasets. This technique estimates the data in a sub-space of its eigenvectors, clusters them in the sub-space, and produces additional samples within the clusters. They also proposed a Twin Neural Network (Twin NN) to learn from large unbalanced datasets further in their works.

Although many recent articles published their datasets of X-rays, they contain up to a few hundred images in the Covid-19

datasets. To build a Covid-19 dataset with enough images in each class, X-rays of confirmed and not confirmed Covid-19 cases must be collected, which are difficult to obtain from reliable and authentic sources. The challenge is further strengthened because the collected data must be correctly annotated [56]. For experimentation purposes, the SARS-CoV-2 CT scan dataset has 1229 Covid-19 samples, whereas the SIRM Covid-19 CT scan and chest X-ray dataset contains just 220 Covid-19 samples. When samples in one class exceed the other, during the CNN and LSTM training, it may be possible to face imbalanced data problems. SMOTE over-sampling on classification models addresses this issue in this analysis. The experimental study with and without SMOTE data augmentation is discussed in Section 4. SMOTE is based on K-nearest neighbor clustering. Euclidean distance calculates K-nearest neighbor for each sample of the minority class. Further, the random nearest neighbor x_i is selected from a bunch of K-nearest neighbors x_j. The copies of new samples are generated by using Eq. (9).

$$x_{new} = x_i + |x_i + x_j| \times \delta \qquad (9)$$

where $\delta \in [0, 1]$ represents a random number between 0 to 1.

3.5. Level 2 − feature extraction

CNN network is quite effective for the processing of high dimensional data such as images and videos etc. However, in the proposed approach, we consider the matrix formed by the handcrafted feature vectors instead of a direct image as an input of the CNN network. Image pixels usually connect locally and globally; similarly, the fused feature vectors represent the locally
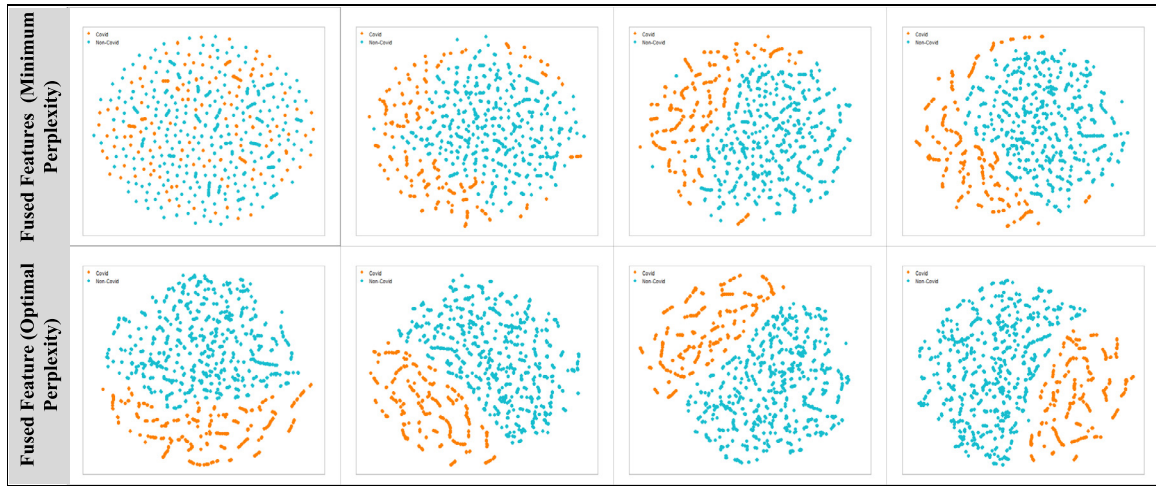
Fig. 4. t-SNE visualization for fused features using minimum (1, 2, 3 and 4) and optimal (5, 6, 7 and 8) perplexity values.

and globally correlated features [27,28]. Therefore, we choose a fused feature set as an input of the CNN network. The fused feature vectors of 1×1024 length greatly help in reducing the training complexity of the CNN network. CNN network extracts more meaningful features with further dimension reduction.

The proposed approach utilizes one dimensional CNN network consisting of two convolution layers, two pooling layers, three dropout layers, and one fully connected layer, as shown in Fig. 5. The CNN network is developed using Python tenser flow library (1.9). The convolution layer filters first slide over the fused feature vectors and then later extract optimal deep features. Features that are derived from each filter are grouped into a new feature set known as a feature map. The optimal length and number of filters are selected by using hyper-parameter tuning. The non-linear activation function ReLU is used on each element as a result. In the proposed CNN network, two convolution layers use 64 and 128 filters, kernel of size four, and padding value same to make the exact shape of output as the input. The spatial size, the number of features, and computational complexity usually decrease using the max-pooling layer. In the proposed CNN network, two max-pooling layers use pooling of size two and strides of size one to obtain a feature map containing the most prominent features of the previous feature map. The proposed CNN network also includes a fully connected classification layer. The overfitting in the proposed CNN network is solved using the Softmax function and dropout layers. The output of the one-dimensional proposed CNN network is expressed by using Eq. (10).

$$o_k^l = f(c_k^l + \sum_{i=1}^{N_{l-1}} Conv1D(X_{ik}^{l-1}, t_i^{l-1}))\tag{10}$$

where c_k^l denotes the scalar bias of the kth neuron at the first layer, the output of ith neuron of layer l-1 is marked with t_i^{l-1}, X_{ik}^{l-1} means kernel weight from the ith neuron at layer l-1 to kth neurons at layer l, and "f()" represents activation function.

3.6. Automatic Covid-19 detection and classification (CNN-LSTM)

The features extracted using max-pooling layers are often passed to the fully connected layer for classification in CNN networks. However, in the proposed CNN network, the sequence of deep features passes to the LSTM layer rather than directly passing them through the fully connected layer for classification. The CNN network efficiently extracts and recognizes the image's local and global structures in the pixel series, while the LSTM network detects long-short-term dependencies [57].

In order to benefit from the characteristics of the two models, the proposed approach introduces a combined CNN-LSTM network for auto-identification and classification of Covid-19. The proposed CNN-LSTM model contains two phases, as shown in Fig. 5. Phase one includes convolution layers and max-pooling layers, whereas phase two consists of the LSTM layer. The local and global information of the fused feature set is encoded by the convolution layers, while the LSTM layer decodes the encoded information. The information is further flattened and passed into a fully connected layer for classification. The workflow of the first phase is covered in Section 3.5.

The LSTM model comprises one memory unit and three other interactive gates: input, forget, and output gates. The state from the previous state is preserved by the memory cell. The input gate defines how much input data from the network must be stored in the unit state at the current time "t". The forget gate determines whether the data will pass or refuse to enter the input gate at time "$t - 1$". The output gate specifies the information for the output. Eq. (11) defines the functionality of the LSTM model.

$$
\begin{aligned}
i_t &= \sigma\,(V_{ix}t + W_i h_{t-1} + b_i)\\
f_t &= \sigma\,(V_{fx}t + W_f h_{t-1} + b_f)\\
\widetilde{c_t} &= \tanh(V_c X_t + W_c h_{t-1} + b_c)\\
c_t &= f_t A c_{t-1} + i_t A\widetilde{c_t}\\
o_t &= \sigma\,(V_o x_t + W_o h_{t-1} + b_o)\\
h_t &= o_t A \tanh(c_t)
\end{aligned}\tag{11}
$$

where x_t denotes the input at time "t", v_* and w_* depicts the weight matrices, b_* and h_* denote the bias and hidden states respectively. σ and tanh represent the activation functions. Input gate, forget gate, output gate, and memory cell are indicated with i_t, f_t, o_t and c_t, respectively.

In Fig. 6, we have introduced two different deep learning models, namely, CNN^2-LSTM and CNN^1-LSTM, respectively. CNN^2-LSTM consists of two convolution layers of 1023 and 512, two max-pooling layers of 512 and 256, one LSTM layer of 128, and a fully connected layer. CNN^1-LSTM consists of one convolution layer of 1023, one max-pooling layer of 512, one LSTM layer of 64, and a fully connected layer.

4. Results and discussions

We chose four evaluation matrices such as precision, recall, f1-score, and accuracy to assess performance. These matrices were measured based on the number of True Positives (TP), False Positive (FP), True Negative (TN), and False Negative (FN) instances,
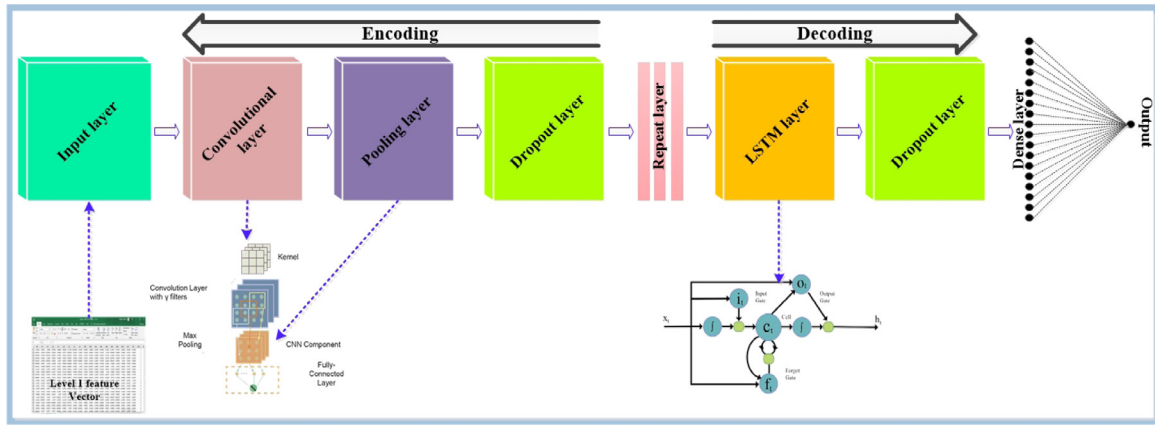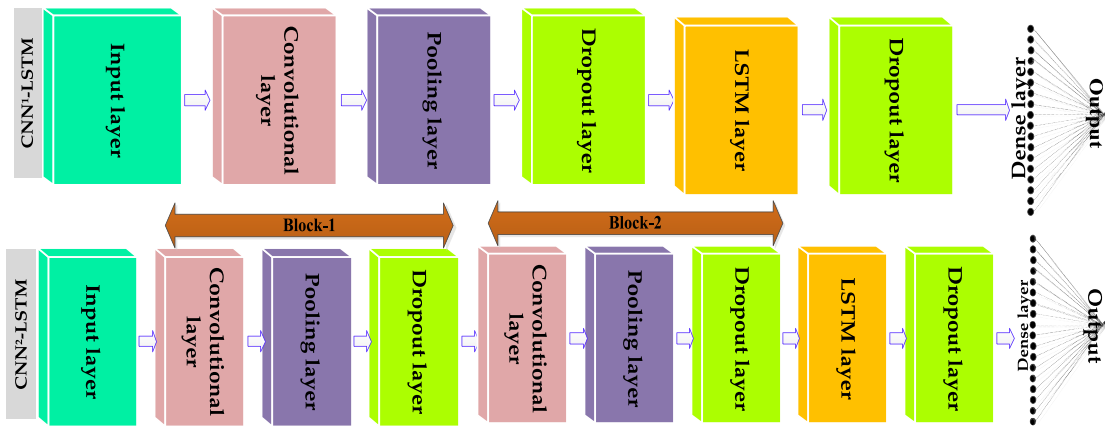
Fig. 5. Proposed CNN-LSTM auto-encoder Network.

Fig. 6. The architecture of proposed CNN^1-LSTM and CNN^2-LSTM.

respectively. The matrices are presented below.

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{FP}{FP + TN}$$

$$\text{F1} - \text{Score} = \frac{2 \times TP}{2 \times TP + FP + FN} \qquad (12)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2492 CT scans from the SARS-CoV-2 dataset (Dataset-1) and 220 CT scans and chest X-rays from the SIRM Covid-19 dataset (Dataset-2) were used for the experiment. We randomly selected 50% of CT scans and chest X-rays for training and testing purposes of each proposed CNN-LSTM model. Table 1 reports detailed layer configurations of both proposed CNN-LSTM networks. These experiments were run to 50 epochs with batch size 32,0.01 learning rate, two fully connected layers, and the sigmoid activation function. The dynamic graphs for accuracy, loss, and receiver operating characteristic (roc) for each proposed CNN-LSTM model are shown in Fig. 7. The blue and red colors display training and testing accuracy lines for both CNN-LSTM models. The green and yellow colors indicate training and testing loss lines for both CNN-LSTM models. Training and testing losses for the CNN^2-LSTM model decrease between 0.68 to 0.02 graph scale. Similarly, the training and testing accuracies for the CNN^2-LSTM model increase between 0.56 to 0.989 graph scale.

Training and testing losses for the CNN^1-LSTM model decrease between 0.70 to 0.14 graph scale. In contrast, the training and testing accuracies for the CNN^1-LSTM model improve from 0.45 to 0.981 graph scale. The accuracy for each proposed CNN-LSTM

model begins from 0.45 graph scale but quickly increases to 0.98 graph scale in only 5 epochs. While the loss for each of the proposed CNN-LSTM models begins from 0.70 graph scale but gradually decreases to 0.02 graph scale in only 50 epochs. The roc curve is formed by drawing the true positive rate (TPR) against the false positive rate (FPR). The true and false positive instances are calculated by using Eq. (12). The roc curves in Fig. 7 appeared closer to the top left corner of the graphs, showing that each proposed CNN-LSTM model obtained better classification efficiency.

Table 2 shows the overall and class-wise performance of the proposed CNN^2-LSTM in both Covid-19 datasets. The percentage of recall indicates correctly classified cases as a particular class in the outcome of both datasets. The results suggest that the true positive ratio for both classes of Dataset-1 was high. As a result, the proposed CNN^2-LSTM had the highest detection rate (98.94%). The mild class had a lower recall percentage (68%) than the other classes in Dataset-2. It was perhaps due to greater confusion in mild and moderate class samples, as illustrated in Fig. 8. The proposed CNN^2-LSTM achieved a maximum classification rate (83.03%) with only 220 Covid-19 samples of Dataset-2. Table 2 shows that the proposed CNN^2-LSTM could assist radiologists and practitioners in detecting Covid-19 and its severity with a lower error rate.

The confusion matrices of both Covid-19 datasets for the proposed CNN^2-LSTM model are presented in Fig. 8. The actual and predicted labels for each class are given vertically and horizontally, respectively. The prediction accuracies of instances were reported in four normalized confusion matrices.

Table 1
Layers' configuration of proposed CNN-LSTM networks.

| Layer (type)	Output shape	Param#
CNN²-LSTM		
conv1d_1 (Conv1D)	(None, 1023, 64)	256
max_pooling1d_1	(None, 512, 64)	0
dropout_1 (Dropout)	(None, 512, 64)	0
conv1d_2 (Conv1D)	(None, 512, 128)	24704
max_pooling1d_2	(None, 256, 128)	0
dropout_2 (Dropout)	(None, 256, 128)	0
lstm_1 (LSTM)	(None, 128)	131584
dropout_3 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 2)	258
Trainable Params: 156,802		
CNN¹-LSTM		
conv1d_1 (Conv1D)	(None, 1023, 64)	256
max_pooling1d_1	(None, 512, 64)	0
dropout_1 (Dropout)	(None, 512, 64)	0
lstm_1 (LSTM)	(None, 64)	33024
dropout_2(Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 2)	130
Trainable Params: 33,410		

CNN¹+LSTM
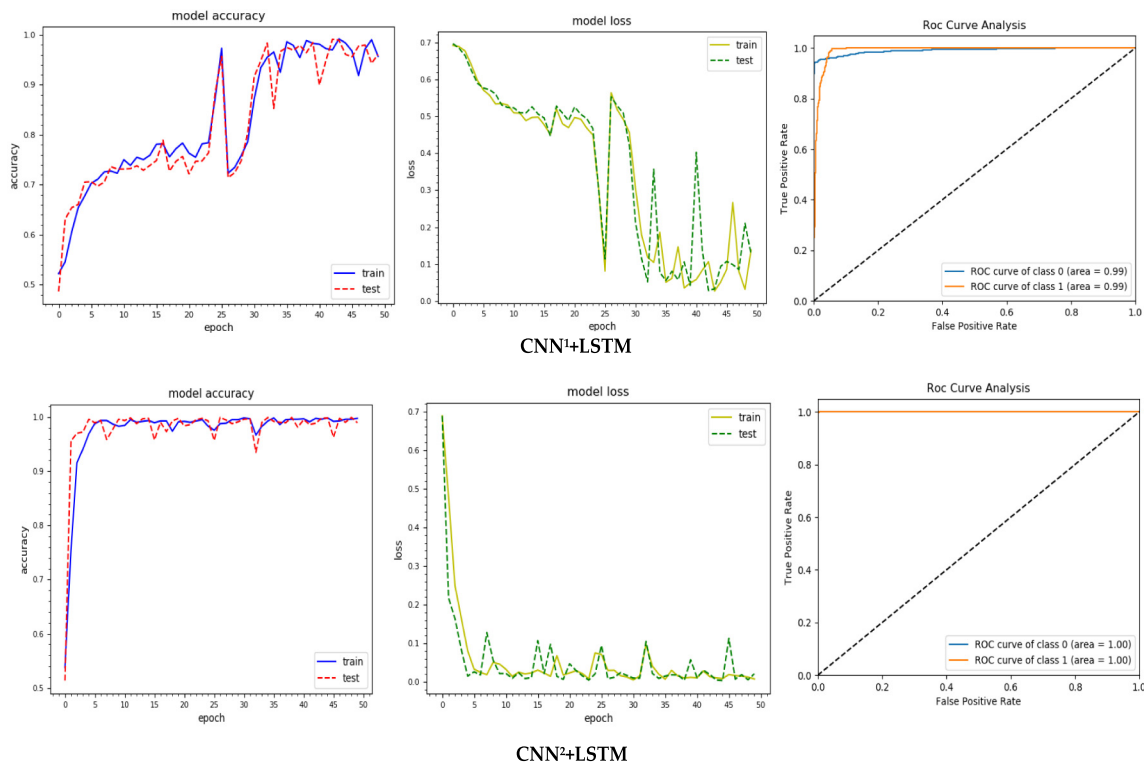
CNN²+LSTM

Fig. 7. Dynamic graphs of each proposed CNN-LSTM model (accuracy, loss, and roc analysis).

The proposed CNN²-LSTM model was tested on imbalanced data at first. Normalized detection and classification rates were displayed in confused matrices. The resulting confusion matrices for the first dataset showed Covid-19 (−) detection rate (0.93), which was lowest in comparison to the Covid-19 (+) detection rate (0.98). It showed that sample distribution in the two classes had been biased. The resulting confusion matrices for the second dataset showed a moderate family classification rate (0.35), which was lowest compared to critical, severe, and mild families' classification rates. Similarly, the classification rate for critical family was 0.4, which was lower than the classification rates for severe and moderate families. The sample ratio between these four families was imbalanced. The moderate family was classified at the highest classification rate (0.84) due to the highest

number of samples. Classifier usually learns more from highest sample class compared to rest classes during training. The overall detection and classification rates were therefore affected. SMOTE subsequently resolved the imbalanced data problem. After using SMOTE, the resulting confusion matrices for the first dataset showed a detection rate of 0.98 for Covid-19 (−) and 1 for Covid-19 (+). The resulting confusion matrices for the second dataset showed a classification rate of 0.87 for moderate family, classification rate of 0.89 for severe family, classification rate of 0.89 for critical family, and classification rate of 0.68 for mild family. The mild family classification rate dropped from 0.84 to 0.68. The few samples in the moderate class showed a pulmonary region with less affected pneumonia anomalies. These samples were

Table 2
Performance analysis of proposed CNN^2-LSTM model for both Covid-19 datasets.

Class wise performance				
Classes	Total	Precision (%)	Recall (%)	F1-score (%)
Dataset-1: SARS-CoV-2 CT scan dataset				
Covid	1229	100	98	99
Non-Covid	1161	98	100	99
Average	–	**99**	**99**	**99**
Dataset-2: SIRM Covid-19 CT scan and Chest X-ray dataset				
Mild	110	85	68	76
Moderate	40	84	87	85
Severe	24	90	89	89
Critical	46	74	89	81
Average	–	**84**	**83**	**83**
Overall performance				
Dataset	**Precision (%)**	**Recall (%)**	**F1-score (%)**	**Accuracy (%)**
Dataset-1	99	99	99	98.94
Dataset-2	84	83	83	83.03

Fig. 8. Confusion matrices of both Covid-19 datasets for proposed CNN^2-LSTM model.

quite similar to mild class samples with fewer pneumonia abnormalities. Therefore, these two classes were more confused in classification. The given results indicate an overall increase in the rates of detection and classification of the proposed CNN^2-LSTM model.

To validate the proficiency of each proposed CNN-LSTM model, we conducted experiments to (1) verify the impact of different training data sizes, (2) ascertain effects of the different number of training layers and parameter sizes on CNN network training efficiency (3) compare the performance of each proposed CNN-LSTM model with state-of-the-art deep learning techniques.

4.1. Verify the impact of different training data size

To assess the impact of various training data sizes on the detection accuracy of different deep learning models, we randomly divided SARS-CoV-2 CT scans into seven different training and test ratios. LSTM, GRU, RNN, DNN, CNN, CNN^1-RNN, CNN^1-GRU, CNN^2-RNN, CNN^2-GRU, CNN^1-LSTM, and CNN^2-LSTM were used for the comparison of detection accuracy, as shown in Table 3.

The highest detection accuracy attained by proposed CNN-LSTM models was 99.74%, CNN-GRU models were 85.36, CNN-RNN models were 98.33%, DNN model was 98%, RNN model was 96.95%, GRU model was 97.73%, and LSTM model was 98.31%, respectively. When the training data split ratios were between 30%–80% of the total dataset, the best values of detection accuracy for proposed CNN-LSTM models were between 98.94%–99.74%. Despite the excellent detection performance of the proposed CNN-LSTM models with data augmentation, the model was still overfitted with few train data split ratios. At 50% train data split ratio, the performance curves of proposed CNN-LSTM models showed stability with no overfitting. As a result, we considered the proposed CNN-LSTM models' performance as a maximum at 50% train data split ratio. In conclusion, the proposed CNN-LSTM

Table 3
Impact of different training data ratios over various deep learning approaches.

Model	30%	40%	50%	60%	70%	80%	90%
LSTM	98	98	96	98.31	93.03	96.95	96.52
GRU	94	94	93	95.47	97.73	93.72	95.21
RNN	89	92	93	92.68	96.95	83.53	88.84
DNN	93.58	98	95.73	97.63	89	96.81	92
CNN^1-RNN	97.63	97.28	96.32	98.19	98.33	96.29	97.35
CNN^1-GRU	76.15	79.50	75.06	74.62	74.72	73.06	68.57
CNN^2-RNN	83.40	97.18	97.74	90.03	96.23	97.54	95.12
CNN^2-GRU	85.36	82.53	82.43	79.71	77.59	79.21	74.06
CNN^1-LSTM	**95.82**	**95.35**	**96.09**	**99.30**	**99.58**	**98.22**	**98**
CNN^2-LSTM	**99**	**98.88**	**98.94**	**99.65**	**99.04**	**99.74**	**98.51**

models outperformed the other deep learning approaches with each training data split ratio.

4.2. Effects of different number of training layers and parameter sizes

To determine the impact of the different number of layers on CNN network training efficiency, we compared the proposed CNN^2-LSTM model with previous works [34–40]. The detection and classification performance was measured based on various indicators such as precision, recall, f-measure, and accuracy. The SARS-CoV-2 Dataset was used in the comparison of Covid-19 detection. As shown in Table 4, the derived fused feature set of the proposed methodology renders the training complexity of the CNN network in contrast to direct input images. The previous work [37] used the pre-trained xDNN, VGG-16, ResNet, AlexNet, and GoogleNet models for Covid-19 detection. They obtained optimal detection accuracy between 97.38% to 91.73%. The previous works [34–36] exhibited the best detection accuracy (96.25%,87.68%, and 99.4%, respectively) with pre-training and layers (121, 18, and 201, respectively). The proposed CNN^2-LSTM model achieved 98.91%∼99.74% detection accuracy without pre-training and 4 layers, as shown in Table 4. Besides this, we also compared the classification performance of the proposed CNN^2-LSTM model with previous works [38–40]. The earlier works [38–40] used Covid-19 images (729, 300, and 408) for severity classification. However, the proposed methodology used only 220 images for Covid-19 severity classification. The previous works [38–40] displayed optimal classification accuracy (95.34%,82%, and 81.9% respectively) with pre-training and layers (201,1 and 34 respectively). Compared to them, the proposed CNN^2-LSTM model achieved 83.03% classification accuracy without pre-training and 5 layers, as shown in Table 4. Compared to previous works [34–40], the fused feature set of the proposed methodology significantly decreased the training complexity of the CNN network.

We compared the proposed CNN-LSTM model with state-of-the-art VGG 16 and VGG 19 models to show the impact of the different number of layers and parameter sizes on network training performance. In Table 5, the performance of models is measured based on their overall detection accuracy, loss, and training time. The proposed CNN-LSTM model used 1024 vectors of fused features as input, whereas the VGG 16 and VGG 19 used full images (200 × 200 dimensions) of the SARS-CoV-2 CT scan dataset input. Compared to conventional VGG 16 and VGG 19 models, which needed entire images as input, the proposed CNN-LSTM model required fewer input parameters and layers for network training (as shown in Table 5). The proposed CNN-LSTM model obtained 99.1% overall detection accuracy, 0.02 loss, and a training time of 182.731 s with only 4 training layers and 156,802 trainable parameters. While traditional VGG 16 and VGG19 models achieved lower overall detection accuracy (49.2% and 54% respectively) and higher loss (0.69,0.69 respectively), this indicates that both models required additional trainable parameters

and training layers for better learning of the whole image input dataset. Overall, the proposed CNN-LSTM model outperforms the conventional VGG 16 and VGG19 models, which take the entire image as input. All experiments were performed at 40 epochs with 16 GB main memory and NVIDIA GeForce RTX 2060 6 Gb GPU.

Fig. 9 shows a brief comparison of training time between the proposed CNN-LSTM model and state-of-the-art deep learning models (VGG 16 and VGG19, respectively). We divided 40 epochs into eight batches to ensure clear graph visibility. The value of each batch indicates the average time of 5 epochs. In the initial 5 epochs, the value of the average training time for CNN-LSTM was high and later gradually declined in the rest of the epochs. The average training time value for both VGG 16 and VGG 19 models in all the 40 epochs was high compared to the proposed CNN-LSTM model. In general, models requiring the entire input image dataset usually take longer to learn more input parameters and layers. The average time given in Fig. 9 shows that the proposed CNN-LSTM model handles sizeable medical datasets with fewer training parameters and layers and less training time.

4.3. Compare the performance of the proposed CNN-LSTM model with state-of-the-art deep learning techniques

We contrasted the detection efficiency of each proposed CNN-LSTM model with the numerous existing deep learning methods such as LSTM, DNN, RNN, GRU, CNN^1-RNN, CNN^1-GRU, CNN^2-RNN, and CNN^2-GRU. Deep learning performance analysis of different methods for Covid-19 detection is shown in Figs. 10, 11,12, and Table 6. We separated 50% of the overall data as training. We trained LSTM with 278,658 parameters, RNN with 69,762 parameters, DNN with 65,666 parameters, GRU with 209,026 parameters, CNN^1-RNN with 8,642 parameters, CNN^2-RNN with 58,114 parameters, CNN^1-GRU with 25,154 parameters, CNN^2-GRU with 123,906 parameters, CNN^2-LSTM with 156,802 parameters, and CNN^1-LSTM with 33,410 parameters. The CNN^2-LSTM contains two convolution layers with lengths 64 and 128, two max-pool-layers with a size of 2, one LSTM layer with length 128, three dropout layers with length 0.1, and one fully connected layer. In comparison, the CNN^1-LSTM contains one convolution layer with length 64, one max-pooling layer with a size of 2, two dropout layers with length 0.1, one LSTM layer with length 128, and one fully connected layer.

In Table 6, we can infer that the proposed CNN^2-LSTM model significantly outperforms the others for precision, recall, f1-score, accuracy, loss, and roc. However, it is impossible to assess the overall detection of deep-learning models with just an accuracy indicator as the models can be over-fit. Therefore, there is a need to observe the loss values of all deep learning models in Table 6, Fig. 7, and Fig. 12. The proposed CNN^1-LSTM obtained excellent performance at the 25th epoch and then decreased performance due to the overfitting of the network, as shown in Fig. 7. The overall loss of CNN^1-LSTM was 0.14. The proposed CNN^2-LSTM

Table 4
Effect of different number of network layers.

Model	Layers	Pre-trained	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
Covid-19 Detection						
Dense [34]	121	Yes	96.29	96.29	96.29	96.25
Voting [35]	18	Yes	NS	NS	NS	87.68
Deep CNNs [36]	18–201	Yes	99.6	NS	66.4	99.4
GoogleNet [37]	NS	Yes	90.20	93.50	91.82	91.73
AlexNet [37]	NS	Yes	94.98	92.28	93.61	93.75
ResNet [37]	NS	Yes	93	97.15	95.03	94.96
VGG-16 [37]	16	Yes	94.02	95.43	94.97	94.96
xDNN [37]	NS	Yes	99.16	95.53	97.31	97.38
CNN^2-LSTM	**4**	**No**	**99**	**99**	**99**	**98.94**
Covid-19 Severity Classification						
Dense [38]	201	Yes	NS	NS	NS	95.34
ANN [39]	1	No	Ns	90	85	82
ResNet-34 [40]	34	Yes	NS	NS	NS	81.9
CNN^2-LSTM	**5**	No	**84**	**83**	**83**	**83.03**

NS = Not specified.

Table 5
Comparison of training performance between proposed CNN-LSTM model and conventional deep learning models.

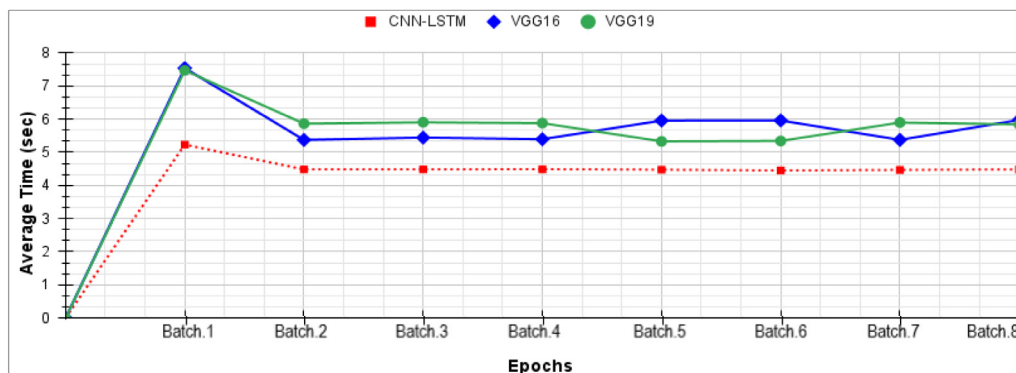| Model	Train parameters	Layers	Accuracy (%)	Loss	Train time (s)
VGG 16	596,738	16	49.2	0.69	235.012
VGG 19	707,522	19	54	0.69	237.58
CNN-LSTM	**156,802**	**4**	**99.1**	**0.02**	**182.731**

Fig. 9. Training time comparison between proposed CNN-LSTM model and conventional deep learning models.

Table 6
Comparison of performance among numerous existing deep learning methods.

Model	Precision (%)	Recall (%)	F1-score (%)	Accuracy %	Loss
LSTM	92	91	91	91.05	0.16
GRU	95	95	95	94.63	0.15
RNN	95	94	94	93.90	0.09
DNN	90	88	87	87.62	1.90
CNN^1-RNN	93	92	92	91.70	0.25
CNN^1-GRU	80	79	79	79.50	0.47
CNN^2-RNN	95	95	95	94.71	0.20
CNN^2-GRU	86	85	85	85.35	0.50
CNN^1-LSTM	**96**	**96**	**96**	**96.09**	**0.14**
CNN^2-LSTM	**99**	**99**	**99**	**98.94**	**0.02**

showed a lower loss of 0.02 with improved predictive capacity than the loss of other deep learning models. Fig. 7 shows that the training and test set errors drop gradually and tend to be constant, demonstrating that the proposed CNN^2-LSTM model is not overfitting. Table 6 indicates that the proposed CNN^2-LSTM achieves good detection performance even with diverse training and test sets. In contrast to the proposed CNN^2-LSTM model, the other deep learning models showed less detection accuracy with different training and test sets, as shown in Fig. 12. As a result, the loss values of these deep learning models in Table 6 were high.

The deep features of the CT scans and X-rays obtained by two convolution layers are distinct from typical time-series data features. The inclusion of more convolution layers and kernels also increases the model's accuracy. The dynamic graphs for accuracy, loss, and receiver operating characteristic (roc) for each proposed CNN-LSTM model and state-of-the-art deep learning techniques are shown in Figs. 7 and 12. The blue and red colors display training and testing accuracy lines for all models. The green and yellow colors indicate training and testing loss lines for all models. In Fig. 7, each proposed CNN-LSTM model's accuracy begins with a 0.45 graph scale but quickly increases to a 0.98 graph scale in only 5–25 epochs. While the loss for each of the proposed
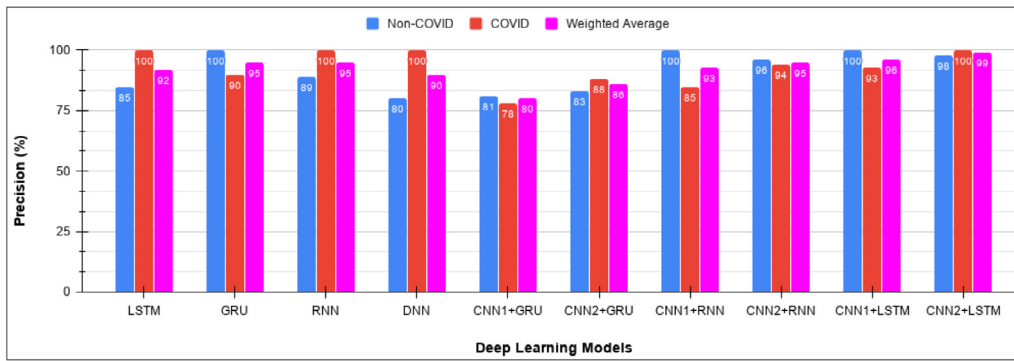
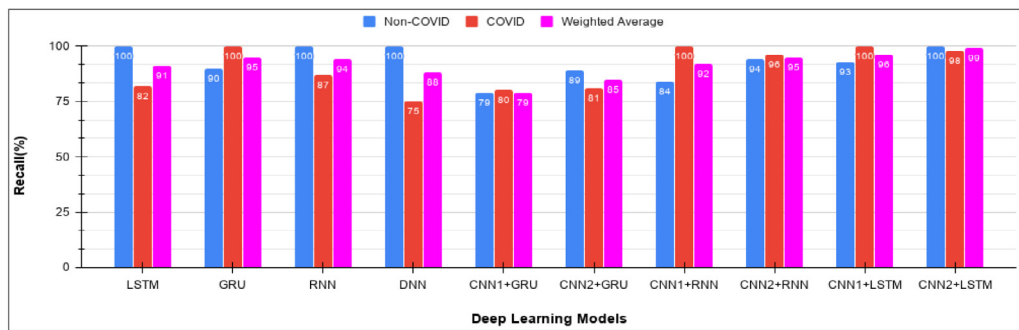**Fig. 10.** Comparison of proposed CNN-LSTM model precision with state of art methods.

Fig. 11. Comparison of proposed CNN-LSTM model recall with state of art methods.

CNN-LSTM models begins from 0.70 graph scale, but gradually decreases to 0.02 graph scale in only 5–25 epochs. The roc curves in Fig. 7 appeared closer to the top left corner of the graphs, showing that each proposed CNN-LSTM model obtained better classification efficiency. Fig. 12 demonstrates overfitting up to 50 epochs with LSTM, DNN, GRU, RNN, CNN^2-RNN, and CNN^1-RNN. The network with a reduced number of layers thus took more epochs to achieve optimum efficiency. CNN^2-GRU and CNN^1-GRU reached optimum efficiency during the 50th epoch, but a low performance was found as these networks were overfitted. Table 5 and Fig. 7 demonstrate that CNN can reliably extract deep characteristics, and LSTM can catch long-term data dependencies significantly from the result. Thus CNN-LSTM layered layout is more sophisticated than traditional deep learning approaches for image recognition. The test data was passed on all the saved models, and we estimated the precision, recall, f1-score, and accuracy as shown in Table 6. The results demonstrated that each proposed CNN-LSTM model significantly outperformed the conventional deep learning techniques in precision, recall, f1-score, and accuracy. Based on precision and recall performance indicators, the detailed comparison of the proposed CNN-LSTM model with state-of-the-art deep learning approaches is provided in Figs. 10 and 11. Figs. 10 and 11 showed that the proposed CNN-LSTM model achieved the best precision, recall, and weighted average value for each class (Covid and Non-Covid), among other deep learning approaches.

5. Conclusions

In this paper, a combined deep learning and multi-level feature extraction methodology were proposed to identify Covid-19 CT scans and chest X-rays. Multi-level feature extraction approach was used to extract features from CT scans and chest X-rays. It significantly improved the training efficiency of the CNN network. The imbalanced data issue was resolved by using the popular SMOTE algorithm. t-SNE visualized the high dimensional features of the level 1 feature description. The LSTM deep learning model was used for Covid-19 detection in the proposed methodology. The Kaggle database's SARS-CoV-2 CT scan dataset and the SIRM Covid-19 CT scan and chest X-ray dataset were employed for experimentation. Experimental outcomes indicated that proposed approach attained 98.94% accuracy with the SARS-CoV-2 CT scan dataset and 83.03% accuracy with the SIRM Covid-19 CT scan and chest X-ray dataset The state-of-the-art deep learning models were used to assess the efficiency of the proposed automated Covid-19 detection approach. The proposed approach has a high success rate with a limited number of Covid-19 samples, which speeds up the treatment of Covid-19 cases. The proposed approach provides help to doctors and radiologists in robust Covid-19 detection and assists them in treating severe cases. There are some limitations to this study that can be further improved in future research. The cross-validation of deep learning models with divergent folds is very important, but the proposed approach randomly separated data due to computational complexities. Moreover, the present research primarily focuses on Covid-19 pneumonia and can be applied to evaluate other types of pneumonia in the future.

CRediT authorship contribution statement

Hamad Naeem: Conceptualization, Methodology, Data curation, Software, Visualization, Investigation, Writing – original draft. **Ali Abdulqader Bin-Salem:** Writing – review & editing.

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.
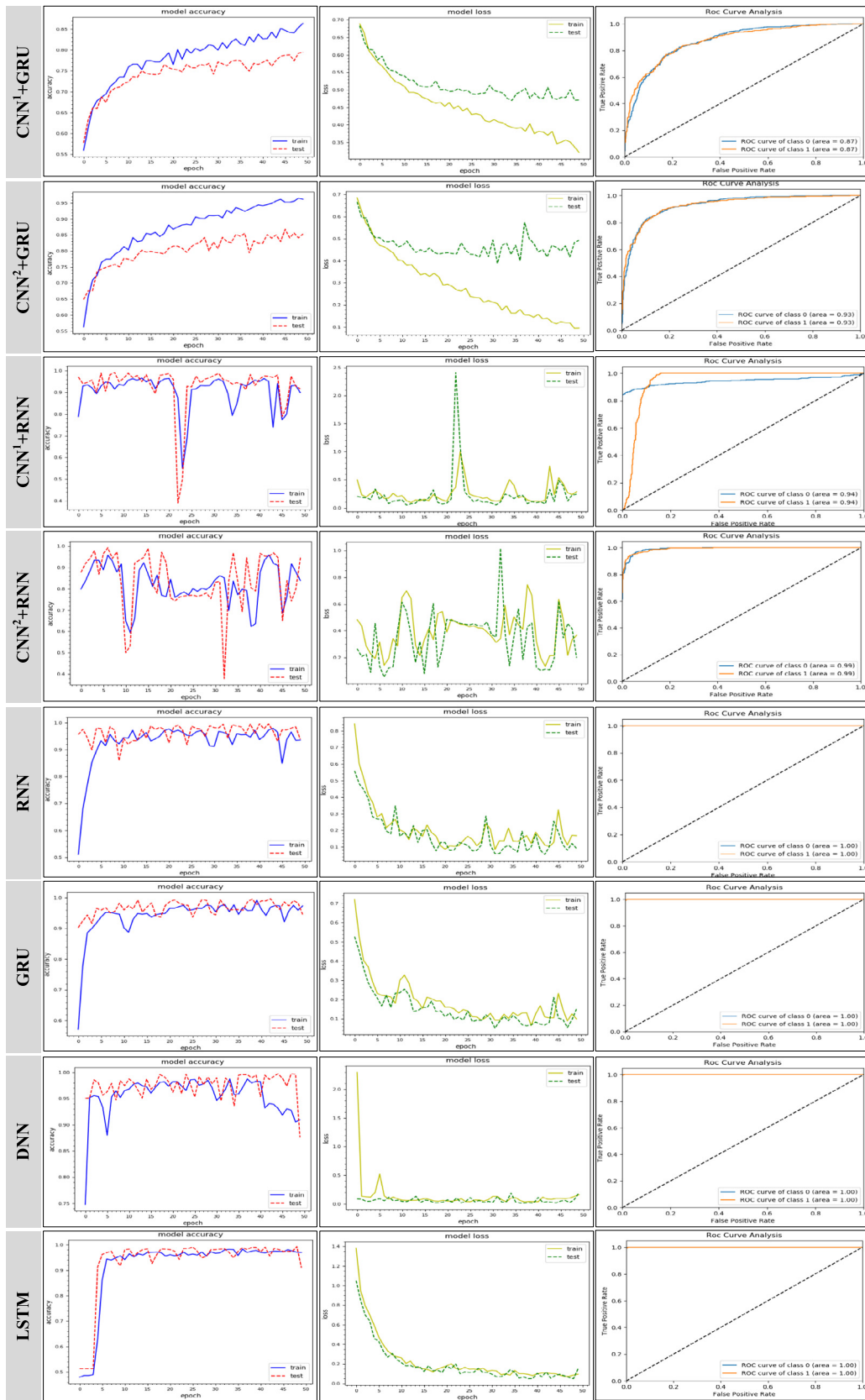
Fig. 12. Dynamic plots of state-of-the art deep learning techniques.

Acknowledgments

References

[1] About world meter Covid-19 data - world meter, 2020, Available online: https://www.worldometers.info/coronavirus/ [accessed on 9 2020].

[2] C. Huang, Y. Wang, X. Li, L. Ren, J. Zhao, Y. Hu, L. Zhang, Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China, Lancet. 395 (2020) 497–506.

[3] Everything about the corona virus- medicine and health, 2020, Available online: https://medicine-and-mental-health.xyz/archieves/4510 [Accessed 6 2020].

[4] W.C. Culp, Coronavirus disease 2019. A & A Pract 2020, 14, e01218.

[5] W. Wang, Y. Xu, R. Gao, R. Lu, K. Han, G. Wu, W. Tan, Detection of SARS-CoV-2 in different types of clinical specimens, JAMA 323 (18) (2020) 1843–1844.

[6] X. Xie, Z. Zhong, W. Zhao, C. Zheng, F. Wang, J. Liu, Chest CT for typical 2019-nCoV pneumonia: Relationship to negative RT-PCR testing, Radiology 296 (2) (2020) E41–E45.

[7] D. Singh, V. Kumar, M. Kaur, Classification of Covid-19 patients from chest CT images using multi-objective differential evolution– based convolutional neural networks, Eur. J. Clin. Microbiol. Infect. Dis. 39 (2020) 1–11.

[8] H.S. Basavegowda, G. Dagnew, Deep learning approach for microarray cancer data classification, CAAI Trans. Intell. Technol. 5 (1) (2020) 22–33.

[9] Y. Tingting, W. Junqian, W. Lintai, X. Yong, Three-stage network for age estimation, CAAI Trans. Intell. Technol. 4 (2) (2019) 122–126.

[10] M. Kaur, H. Kumar Gianey, D. Singh, M. Sabharwal, Multiobjective differential evolution based random forest for e-health applications, Modern Phys. Lett. B 33 (05) (2019) 1950022.

[11] M. Hammad, A. Maher, K. Wang, F. Jiang, M. Amrani, Detection of abnormal heart conditions based on characteristics of ECG signals, Measurement 125 (2018) 634–644.

[12] A.S. Alghamdi, K. Polat, A. Alghoson, A.A. Alshdadi, A.A. Abd El-Latif, A novel blood pressure estimation method based on the classification of oscillometer waveforms using machine-learning methods, Appl. Acoust. 164 (2020) 107279.

[13] H. Khalil, N. El-Hag, A. Sedik, W. El-Shafie, A.E.N. Mohamed, A.A. Khalaf, A.S. El-Fishway, Classification of diabetic retinopathy types based on convolution neural network (CNN), Menoufia J. Electr. Eng. Res. 28 (2019) 126–153.

[14] N.T. Haggag, A. Sedik, G.M. Elbanby, A.S. El-Fishawy, A.A. Khalaf, Classification of corneal pattern based on convolutional LSTM neural network, Menoufia J. Electr. Eng. Res. 28 (2019) 158–162.

[15] J. Thevenot, M.B. Lopez, A.A. Hadid, Survey on computer vision for assistive medical diagnosis from faces, IEEE J. Biomed. Health Inf. 22 (2018) 1497–1511.

[16] M.M. Islam, H. Iqbal, M.R. Haque, M.K. Hasan, Prediction of breast cancer using support vector machine and K-Nearest neighbors, in: 2017 IEEE Reg. 10 Humanit. Technology Conference IEEE, 2017, pp. 226–229.

[17] M.R. Haque, M.M. Islam, H. Iqbal, M.S. Reza, M. Hasan, Performance evaluation of random forests and artificial neural networks for the classification of liver disorder, in: International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2), 2018, pp. 1–5.

[18] M.K. Hasan, M.M. Islam, M.M.A. Hashem, Mathematical model development to detect breast cancer using multigene genetic programming, in: 5th International Conference on Informatics, Electronics and Vision (ICIEV), 2016, pp. 574–579.

[19] Islam, S. Ayon, I.M. Milon, Diabetes prediction: a deep learning approach, Int. J. Inf. Eng. Electron. Bus. 11 (2019) 21–27.

[20] S.I. Ayon, M.M. Islam, M.R. Hossain, Coronary artery heart disease prediction: a comparative study of computational intelligence techniques, IETE J. Res. (2020) 1–20.

[21] A. Rahaman, M. Islam, M. Islam, M. Sadi, S. Nooruddin, Developing IoT based smart health monitoring systems: a review, Revue Intell. Artif. 33 (2019) 435–440.

[22] M.M. Islam, A. Rahaman, M.R. Islam, Development of smart healthcare monitoring system in IoT environment, SN Comput. Sci. 1 (2020) 185.

[23] X. Jiang, Feature extraction for image recognition and computer vision, in: 2nd IEEE International Conference Computer Science Information Technology ICCSIT, 2009, pp. 1–15.

[24] S.K. Lakshmanaprabu, S.N. Mohanty, K. Shankar, N. Arunkumar, G. Ramirez, Optimal deep learning model for classification of lung cancer on CT images, Future Gener. Comput. Syst. 92 (2019) 374–382.

[25] Y. Chen, X. Gou, X. Feng, Y. Liu, G. Qin, Q. Feng, W. Yang, W. Chen, Bone suppression of chest radiographs with cascaded convolutional networks in wavelet domain, IEEE Access 7 (2019) 8346–8357.

[26] A. Sedik, A.M. Iliyasu, B. Abd El-Rahiem, M.E. Abdel Samea, A. Abdel-Raheem, M. Hammad, J. Peng, F.E. Abd El-Samie, A.A. Abd El-Latif, Deploying machine and deep learning models for efficient data-augmented detection of Covid-19 infections, Viruses 12 (2020) 769.

[27] A.M.X. Elmoogy, T.L.R. Dong, K.R. Westendorp, SurfCNN: A descriptor enhanced convolutional neural network, in: IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), Victoria, BC, Canada, 2019, pp. 1-5.

[28] A.M. Elmoogy, X. Dong, T. Lu, R. Westendorp, K.R. Tarimala, Surfcnn: A descriptor accelerated convolutional neural network for image-based indoor localization, IEEE Access 8 (2020) 59750–59759.

[29] L. Sarker, M. Islam, T. Hannan, Z. Ahmed, Covid-DenseNet: A deep learning architecture to detect Covid-19 from chest radiology images, 2020, Preprint Available online: https://10.20944/preprints202005.0151.v [accessed on 9 2020].

[30] F. Shan, Y. Gao, J. Wang, W. Shi, N. Shi, M. Han, Z. Xue, D. Shen, Y. Shi, Lung infection quantification of Covid-19 in CT images with deep learning, 2020.

[31] J. Zhang, Y. Xie, Y. Li, C. Shen, Y. Xia, Covid-19 screening on chest x-ray images using deep learning-based anomaly detection, 2020.

[32] S. Wang, Y. Zha, W. Li, Q. Wu, X. Li, M. Niu, M. Wang, X. Qiu, H. Li, H. Yu, W. Gong, A fully automatic deep learning system for Covid-19 diagnostic and prognostic analysis, Eur. Respir. J. 56 (2020) 2000775.

[33] P.K. Sethy, S.K. Behera, Detection of coronavirus disease (Covid-19) based on deep features, 2020, Preprint Available online: http://arxiv.org/abs/2003.11055 [accessed on 9 2020].

[34] J. Aayush, G. Neha, S. Dilbag, K. Vijay, K. Manjit, Classification of the Covid-19 infected patients using DenseNet201 based deep transfer learning, J. Biomol. Struct. Dyn. (2020) 1–8.

[35] P. Silva, E. Luz, G. Silva, G. Moreira, R. Silva, D. Lucio, D. Menotti, Covid-19 detection in CT images with deep learning: A voting-based scheme and cross-datasets analysis, Inform. Med. Unlocked 20 (2020) 100427.

[36] H. Alshazly, C. Linse, E. Barth, T. Martinetz, Explainable Covid-19 detection using chest CT scans and deep learning, Sensors 21 (2021) 455.

[37] S. Angelov, M. Biaso, F. Higa, K. Daniel, SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification eduardo soares, 2020, medRxiv 2020.04.24.20078584.

[38] Z. Yu, Rapid identification of COVID-19 severity in CT scans through classification of deep features, Biomed. Eng. 19 (1) (2020) 1–13.

[39] A.R. Carvalho, COVID-19 chest computed tomography to stratify severity and disease extension by artificial neural network computer-aided diagnosis, Front. Med. 7 (12) (2020) 1–11.

[40] L.S. Xiao, et al., Development and validation of a deep learning-based model using computed tomography imaging for predicting disease severity of coronavirus disease, Front. Bioeng. Biotechnol. 8 (7) (2020) 1–11.

[41] National Health Commission, Diagnosis and treatment protocol for novel coronavirus pneumonia (Trial Version 7), Chin. Med. J. 133 (2020) 1087–1095.

[42] M.A. Warren, Z. Zhao, T. Koyama, Severity scoring of lungs edema on the chest radiograph is associated with clinical outcomes in ARDS, Thorax 73 (9) (2018) 840–846.

[43] J. Hernandez, A. C-. Ochoa, J.F.M. Trinidad, An empirical study of oversampling and under sampling for instance selection methods on imbalance datasets, in: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (Lecture Notes in Computer Science), Vol. 8258, 2013, pp. 262–269.

[44] D. Lowe, Object recognition from local scale-invariant features, ICCV (1999) 1150–1157.

[45] F.L. Fei, P. Perona, A Bayesian hierarchical model for learning natural scene categories, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 524–531.

[46] H. Bay, A. Ess, T. Tuytelaars, G.L. Van, Speeded-up robust features (SURF), Comput. Vis. Image Underst. (2008) 346–359.

[47] N. Hamad, G. Bing, R.N. Muhammad, U. Farhan, A. Hamzah, S.J. Muhammad, Identification of malicious code variants based on image visualization, Comput. Electr. Eng. 76 (2019) 225–237.

[48] F. Ciompi, C. Jacobs, E.T. Scholten, et al., Bag-of-frequencies: A descriptor of pulmonary nodules in computed tomography images, IEEE Trans. Med. Imaging (2015) 962–973.

[49] Q. Tian, S. Zhang, Descriptive visual word sand visual phrases for image applications, ACM Multimedia (2009) 19–24.

[50] K. Yan, R. Sukthankar, PCA-SIFT: A more distinctive representation for local image descriptors, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2004, pp. 506–513.

[51] Hamad, Detection of malicious activities in internet of things environment based on visualization images and machine intelligence, in: Wireless Personal Communications, Springer, 2019, pp. 1–21.

[52] L.v.d. Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008) 2579—2605.

[53] P.Z. Salam, S. Najafi, Evaluating Prediction Accuracy for Collaborative Filtering Algorithms in Recommender Systems. Diva-Portal.Org, Royal Institute of Technology, 2016, Preprint Available online https://10.20944/preprints202005.0151.v [accessed on 28 2021].

[54] Jayadeva, S. Soman, Saxena, Eigensample: A non-iterative technique for adding samples to small datasets, Appl. Soft Comput. 70 (2018) 1064–1077.

[55] Jayadeva, H. Pant, M. Sharma, S. Soman, Twin neural networks for the classification of large unbalanced datasets, Neurocomputing 343 (2019) 34–49.

[56] S. Rahman, S. Sarker, M.A. Al Miraj, R.A. Nihal, A.N. Haque, A. Al Noman, Deep learning–driven automated detection of Covid-19 from radiography images: A comparative analysis, Cogn. Comput. (2021) 1–30.

[57] J. Yang, J. Qu, Q. Mi, Q. Li, A CNN-LSTM model for tailings dam risk prediction, IEEE Access 8 (2020) 206491-206502.