I.J. Image, Graphics and Signal Processing, **2025, 5, 76-97**
Published Online on October 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2025.05.06

Deep Learning based Triple Task Learning Framework for COVID-19 Severity Detection Using CT Images

Krishna Bhimaavarapu
Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram 522 502, Andhra Pradesh, India
E-mail: bkrishna@kluniversity.in
ORCID ID: https//orcid.org//0000-0003-4833-1100

Amarendra K.*
Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram 522 502, Andhra Pradesh, India
E-mail: amarendra@kluniversity.in
ORCID ID: https//orcid.org//0000-0003-3597-0878

Received: 29 January, 2024; Revised: 15 April, 2024; Accepted: 19 April, 2025; Published: 08 October, 2025

Abstract: The new form of coronavirus started in Wuhan, China, in 2019 and is known as COVID-19. It created severe health issues and also deaths in most of the countries. The test kits and certain imaging techniques, namely computed tomography and X-ray, are utilized to analyze the severity of diseases. Earlier, researchers introduced several machine-learning techniques for medical diagnosis. However, due to complexity concerns and a high error rate, such strategies cannot produce superior results. Recently, several deep learning mechanisms have been utilized in medical diagnosis. In this work, a new triple-task learning architecture is introduced for the identification and categorization of COVID-19 disease by referring to CT images. First, the input images are pre-processed utilizing Gabor filtering and image resizing. After pre-processing, the images are fed to the triple-task learning network. Here, in the proposed network, three modules are included, namely Residual Swin Transformer based U-Net, Deep convolution and Extended BiLSTM. In this, the Residual Swin Transformer-based U-Net performs the segmentation task. After that, the most significant features are extracted using Deep convolution. The extracted features are then used in the classification step when the various classes of COVID-19 are classified. Finally, the classification parameters are fine-tuned utilizing the Adaptive Fire Hawks algorithm. Then, the proposed technique is experimentally verified utilizing a Python tool, and the performance is analyzed by evaluating the performance metrics. Also, the proposed approach is compared to existing techniques, and the comparison results show that the proposed technique achieves better performance, having an accuracy of 99.46%.

Index Terms: Gabor Filter, Residual Swin Transformer Based U-Net, Deep Convolution, Extended Bilstm, Fire Hawks Algorithm

1. Introduction

COVID-19, a new coronavirus discovered in China's Wuhan in December 2019, infects the lungs and causes pneumonia [1]. Despite the Chinese government's efforts, such as frequent hand washing, wearing masks in public places, and practicing social distancing, the COVID-19 outbreak spread quickly and widely [2]. COVID-19 should be deemed a public health emergency, according to the World Health Organization. Later, on March 11, 2020, WHO labeled COVID-19 a pandemic, and it has since spread over the world. The WHO has given guidelines to governments, individuals, and multinational and national corporations to check the entire containment of the virus [3]. The number of cases infected by this disease has rapidly increased, and preventive and treatment measures must be implemented to manage the disease and save people's lives. The incubation period of the virus is long, and it takes over two weeks to show symptoms, or in some cases, it will not show any symptoms. This is one of the reasons for the rapid spreading of the disease. As a result, detecting and quarantining COVID-19-positive cases as soon as possible will save many lives

[4, 5]. This disease easily spreads through physical contact, air and also hand contact with persons who are infected. The virus inserts inside the lung cells via the respiratory system, and there it replicates.

COVID-19 is made up of ribonucleic acid (RNA), making it difficult to diagnose and treat the disease due to its mutational behaviour. Common COVID-19 symptoms include cough, fever, dizziness, muscle aches, difficulty breathing, and headaches [6]. The disease is very dangerous because it is a new disease, and researchers cannot completely understand the disease. The virus is very harmful, and it weakens the immune system, thereby causing the death of the people. Physicians and specialists all over the world are looking to develop a treatment for a particular disease [7]. A test referred to as reverse transcription polymerase chain reaction (RTPCR) is taken to detect persons who are positive for COVID-19. Besides this, RTPCR X-ray findings and CT images are other tools for diagnosing COVID-19. Physicians check the lung images and identify the signs of deformation on the X-ray images due to COVID-19 [8, 9]. However, this procedure needs some time to differentiate correctly among several types of coronavirus, namely, SARS, COVID-19, and MERS. So, physicians are searching for new and fast-leading techniques for monitoring, tracking and restricting the COVID-19 disease. One of the important limitations faced by COVID-19 research is the lack of adequate and accurate results [10].

One of the technologies that helps physicians with population screening, notification, medical support, and infection control recommendations is artificial intelligence [11]. Doctors can utilize artificial intelligence techniques on CT scans as well as X-rays for analysis. This technology will also be used for drug screening in order to identify treatments that can fight the COVID-19 disease. Artificial intelligence is an improved tool that can help with planning, drug development, and recommending follow-ups for COVID-19 patients [12, 13]. Computer intelligence refers to machine learning (ML), a subfield of artificial intelligence [14-17]. Deep learning (DL) is another Machine Learning-based technique for better learning of image features. Deep learning has the primary advantage of being able to learn from vast volumes of data [18]. DL approaches have recently been widely used in the diagnosis and detection of COVID-19 disease [19, 20]. Deep learning is a type of computer software that uses neural networks to learn visual information. A new multi-task deep learning technique for illness classification and segmentation is proposed in this research.

Motivation: The coronavirus (COVID-19) has been declared a medical emergency by the World Health Organization. The new coronavirus illness called COVID-19 is spreading more quickly day-to-day. Controlling and curing this illness is crucial, and physicians may benefit greatly from any scientific device that helps identify coronavirus quickly. In this setting, cutting-edge automation such as image processing, machine learning, and CT and chest radiography (CXR) are developed as a potentially effective countermeasure against COVID-19. There are significantly more accurate and helpful approaches to classify COVID-19 within the epidemic zone, such as RT-PCR and CT imaging of the chest. CT scanners are present in most hospitals, and the utilization of chest CT scans for patient categorization and early diagnosis will be beneficial. CT is one of the most effective modalities for detecting infections in patients. Recently, deep learning techniques are also utilized for detecting the COVID-19 disease. Such techniques can deeply learn the features and help for better classification. Hence, in the proposed work, a new deep learning-dependent Triple Task Learning Framework is utilized for the severity detection as well as categorization of COVID-19 disease. The major objectives of the proposed work are,

- To introduce a new triple-task learning framework for enhancing the efficiency in COVID-19 severity classification.
- To pre-process the input images using Gabor filtering and image resizing to reduce the unwanted noises in the inputs.
- To introduce a Residual Swin transformer-based U-Net model in the first module for segmenting the affected regions from the chest CT scans.
- Utilize a deep convolutional network in the second module to enable feature extraction tasks.
- Based on the learned features, the presence of COVID-19 severities is categorized by utilizing an extended BiLSTM in the final module.
- To demonstrate the strength of the suggested work, examine the performance of the proposed study using multiple criteria and compare the results to existing approaches.

The remaining sections of the paper are organized as follows: Section 2 explains the related works based on recently existing methodologies, Section 3 presents the proposed methodology in detail, Section 4 describes the results and discussion, and Section 5 presents the conclusion and future scope.

2. Related Works

Kogilavani et al. [21] used deep learning algorithms to identify COVID-19 based on Lung CT images. Deep learning is utilized to identify between healthy and COVID-19 patients using computed tomography (CT) of patients. CNN architectures such as DenseNet121, VGG16, MobileNet, Xception, NASNet, and EfficientNet are used in this work. The dataset includes 3873 CT scans with and without COVID. The dataset was divided into three parts: training, validation, and testing. Here, the classification output shows that VGG16 gives better results than the other architectures.

Gaurav et al. [22] suggested an X-ray-based deep learning and optimization-based technique for assessing COVID-19 illness, a new coronavirus. In this work, multi-objective deep learning as well as optimization techniques are utilized to identify coronavirus-affected patients. The J48 decision tree algorithm detects deep features in corona-affected X-ray images. To detect infected patients, 11 CNN architectures are used in this study: VGG16, Alexnet, GoogleNet, ResNet101, ResNet 50, ResNet18, InceptionV3, DenseNet201, InceptionResNetV2 and XceptionNet. The efficacy is expressed using the k-fold validation technique. The parameters are optimized using a multi-objective spotted hyena optimizer (MOSHO).

Deep learning architectures were introduced by Chouat et al. [23] to diagnose COVID-19 in CT and CXR images. The ability of deep transfer learning is identified in this study by building a classifier for recognizing COVID-19-positive cases using CXR and CT images. Data augmentation is utilized in this scenario to increase the dataset size to avoid overfitting and improve generalization capabilities. A series of deep networks are utilized, namely Inception V3, Xception, ResNet50, VGG 19 and InceptionV3. From the results, it can be observed that VGG 19 is better than other networks.

Demir et al. [24] suggested detecting COVID-19 disease using a robust and effective deep-learning technique. In this study, a new deep learning algorithm called DEepCOvNet is utilized to classify X-rays of the chest into normal, COVID-19, and pneumonia classes. Here, the convolutional autoencoder consists of convolution units in the decoder as well as encoder blocks. Features were selected using an efficient and new algorithm called sequentially discounting auto regressive (SDAR). For classification, SVM is utilized along with various kernels. In addition, the hyperparameters are optimized with the help of the Bayesian algorithm.

Afif et al. [25] introduced a deep learning-dependent approach for segmentation of lesions in CT scan images for COVID-19 detection. Here, a new deep-learning mechanism is suggested for COVID-19 analysis and segmentation. This system is built on a neural network that is contextually aggregated. It comprises three components: the attention mix module (AMM), the context fuse module (CFM), and the residual convolutional module (RCM). It can detect consolidation area and ground class opacity. These lesions commonly refer to general pneumonia as well as COVID-19 patients.

Problem Statement: COVID-19 infects people all over the world, and as the number of affected people increases, it is critical to monitor and categorize both infected and healthy persons. Also, proper treatment is necessary for the affected people. Previously, researchers introduced approaches such as RT-PCR, machine learning, CT, and chest radiography for the diagnosis of COVID-19. Alternative alternatives are presented because the moratorium period has a strong association with test results and large erroneous negative estimations. Several recent works investigate deep learning models used for categorization in medical diagnosis utilizing Computed Tomography (CT) data. However, some of the existing techniques are ineffective due to increased complexity and high error rates. As a result, for the correct identification and categorization of COVID-19, an effective deep-learning network is required.

3. Proposed Methodology

COVID-19 infection analysis is a challenging task due to the intricacy of several learning methods. Thus, the goal of this study is to develop an effective multi-task learning framework that combines transformer-based U-Net segmentation with an integrated deep-learning mechanism for detecting COVID-19 disease in CT images. Fig.1 depicts the proposed work's workflow.

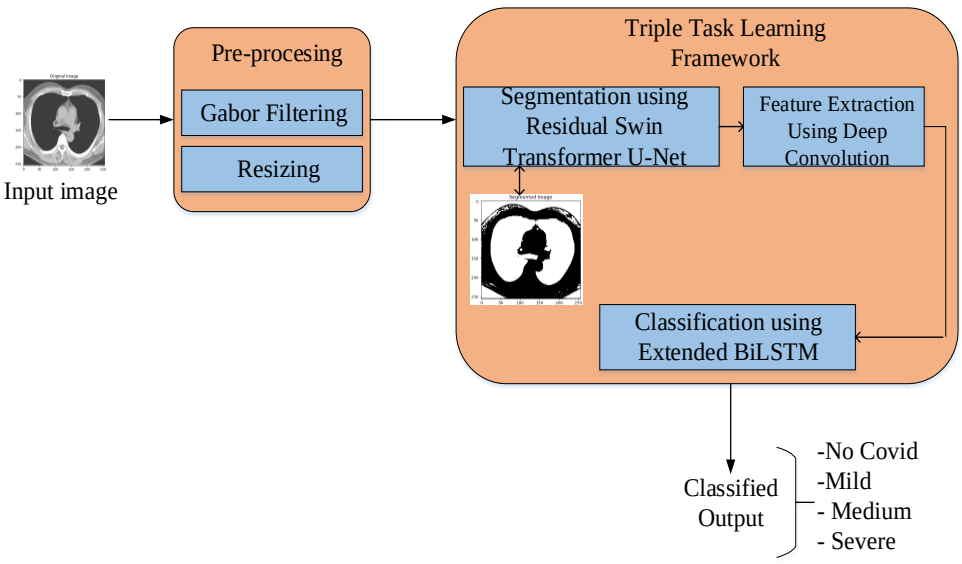


Fig. 1. Workflow of proposed work

Pre-processing, segmentation, feature extraction, and classification are the most important stages in this proposed study. The incoming images are first pre-processed with Gabor filtering and image cropping. As input, these pre-processed photos are fed into the proposed Triple-Task Learning Framework. The following modules are included in the proposed framework: Residual Swin Transformer-based U-Net, Deep Convolution, and Extended BiLSTM. Here, module 1 performs a segmentation task; module 2 is responsible for extracting the features, and module 3 is used for enabling classification tasks using the softmax layer. Finally, an Adaptive Fire Hawks Algorithm (AFHA) is used to fine-tune the proposed technique's hyper-parameters. As a result, the proposed classification stage classifies the severity of COVID-19 disease based on the inputs.

3.1 Pre-processing

The input is first pre-processed using Gabor filtering and image resizing. Gabor filtering is used in pre-processing to remove noise, and input images are resized to the same size using image resizing.

3.1.1 Gabor filtering

A Gabor filter is a type of linear filter in which the impulse response is determined by a sinusoidal wave convoluted with a Gaussian function. Gabor filtering is a common approach for texture feature analysis, and this filter can perform trigonometric convolution on the Gaussian function. Various scales and features from the input image are detected by selecting the appropriate Gabor function. As a result, the Gabor filter is used in edge detection and denoising applications. The Gabor filter is used for image denoising in this work. The Gabor filter is used to denoise an image in Fig.2.

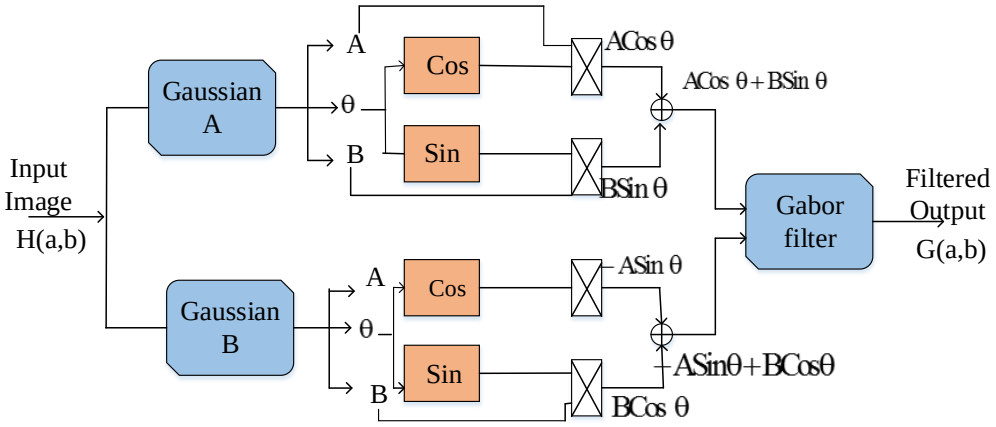


Fig. 2. Noise removal using Gabor filter

When the statistical equations for the Gabor filter are addressed, the 2D- Gabor filter is demonstrated as in the following equation (1).

$$F(a,b) = R(a,b) * V(a,b) \tag{1}$$

Here, equation (1) specifies the convolution among $R(a,b)$ and $V(a,b)$ which are provided in equation (2) and (3).

$$R(a,b) = e^{-j2\pi(z_0 a+\upsilon)} \tag{2}$$

$$V(a,b) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{a^2}{\sigma_a^2}+\frac{b^2}{\sigma_b^2}\right)} \tag{3}$$

The equations (1) will be expressed as in equation (4).

$$F(a,b) = e^{-\frac{1}{2}\left(\frac{a^2}{\sigma_a^2}+\frac{b^2}{\sigma_b^2}\right)\cdot e^{-j2\pi(z_0 a+v_0 b)}} \tag{4}$$

This equation (4) will gain simplified and given in equation (5)

$$F(a,b;\sigma,\phi,\xi,\beta,\theta) = e^{-\frac{1}{2}\left(\frac{a'^2+}{\sigma^2}\right)} e^{i\left(2\pi\frac{a'}{\beta}+\phi\right)} \tag{5}$$

From equation (5),

$$\mathrm{Re}\{F(a,b;\sigma,\phi,\xi,\beta,\theta)\} = e^{-\frac{1}{2}\left(\frac{a'^2+\xi^2 b'^2}{}\right)}\cos\left(2\pi\frac{a^1}{\xi}+\phi\right) \tag{6}$$

$$\mathrm{Im}\{a,b;\sigma,\phi,\xi,\beta,\theta\} = e^{-\frac{1}{2}\left(\frac{a'^2+\xi^2 b'^2}{\sigma^2}\right)}\sin\left(2\pi\frac{a'}{\beta}+\phi\right) \tag{7}$$

Here, ϕ specifies the Gabor kernel function's phase parameter, β represents the wavelength, β value can be greater than or equal to 2, it cannot be greater than $1/5^{th}$ of the size of the input image. The image will be more clear if β attains a greater value.

Here, the ratio $\frac{\sigma}{\beta}$ denotes the half response bandwidth for the spatial frequency of the filter, which varies among the lower and upper frequencies. Here, in equations (8) and (9), σ represents the Gaussian factor standard deviation of the Gabor filter.

$$BW(y) = \log_2^{\frac{\frac{\sigma}{\beta}\cdot\pi+\sqrt{\frac{\ln 2}{2}}}{\frac{\sigma}{\beta}\cdot\pi-\sqrt{\frac{\ln 2}{2}}}} \tag{8}$$

$$\frac{\sigma}{\beta} = \frac{1}{\pi}\cdot\sqrt{\frac{\ln 2}{2}}\cdot\frac{2^y+1}{2^y-1} \tag{9}$$

The value σ differs with bandwidth β and is fixed simultaneously. The standard deviation will be specified with respect to wavelength. Here, the RGB image is transformed to greyscale during the pre-processing phase. Here, a pixel having 8 bits is selected, and the quantization levels are from zero to 255. In this study, 256x256 size for the pixel is taken as input. After that, the grey image is converted to a binary image. The image thus attained is $H(a,b)$ and here, a and b are the spatial coordinates. The input image $H(a,b)$ is subjected to convolution along with the Gabor kernel coefficient $fy(a,b)$ to obtain a filtered image $G(a,b)$.

3.1.2 Image resizing

The size of the input image generated by the dataset may vary. As a result, all of the images must be resized to the same size. The input images in this work are of various sizes, and they are scaled to 256x256.

3.2 Segmentation, Feature extraction and Classification using Triple-Task learning Framework

A triple-task learning architecture is used in this study to recognize COVID-19 in CT images. The triple task learning network contains a Residual Swin transformer-based U-Net for segmentation. Also, the network has a deep convolution network for feature extraction. In addition, it consists of an extended BiLSTM for classifying the disease. The triple-task learning framework is illustrated in Fig.3.

3.2.1 Segmentation utilizing Residual Swin transformer based U-Net

Here, the residual U-Net used in this work comprises an encoder-decoder architecture. *Encoder:* The residual U-Net encoder portion is a contraction path designed to collect contextual semantic information, and it is an upgraded version of the actual U-Net coupled with Resnet. The max pooling unit and 7x7 convolution unit at the front of ResNet are removed, but the 3x3 convolution unit at the beginning of U-Net is retained to increase the number of channels from 3 to 64. In the actual U-Net, after performing 4 downsampling, the amount of channels will be 1024. For minimizing the computational complexity and also model parameters, the resulting amount of channels in residual U-Net is 512. This is achieved after performing four downsampling operations. If four downsamplings are finished, the feature image's resolution will be changed to $1/16^{th}$ of the actual image.

The decoder portion of residual U-Net is an elaborated path, where the feature samples are up-sampled, and the resolution of the feature map is enhanced. Using skip connection, the obtained feature map after each upsampling was linked with the feature map of the associated downsampling path. Here, the skip connection mechanism reuses the eliminated details of the image in the encoding. Hence, the decoding unit can restore the information most efficiently.

Swin Transformer Block: The standard transformer contains P similar blocks, and every block comprises of Multi-head self attention (MSA) as well as Multi-Layer Perceptron (MLP). In addition, a layer norm (LN) block is used before

each MSA module and MLP, and a residual link is used after each module. Hence, the result of I_p of p -layer in the encoder of the transformer is written as:

$$\hat{I}_p = MSA\left(LN\left(z_{p-1}\right)\right) + I_{p-1} \tag{10}$$

$$I_p = MLP\left(LN\left(\hat{I}_p\right)\right) + \hat{I}_p \tag{11}$$

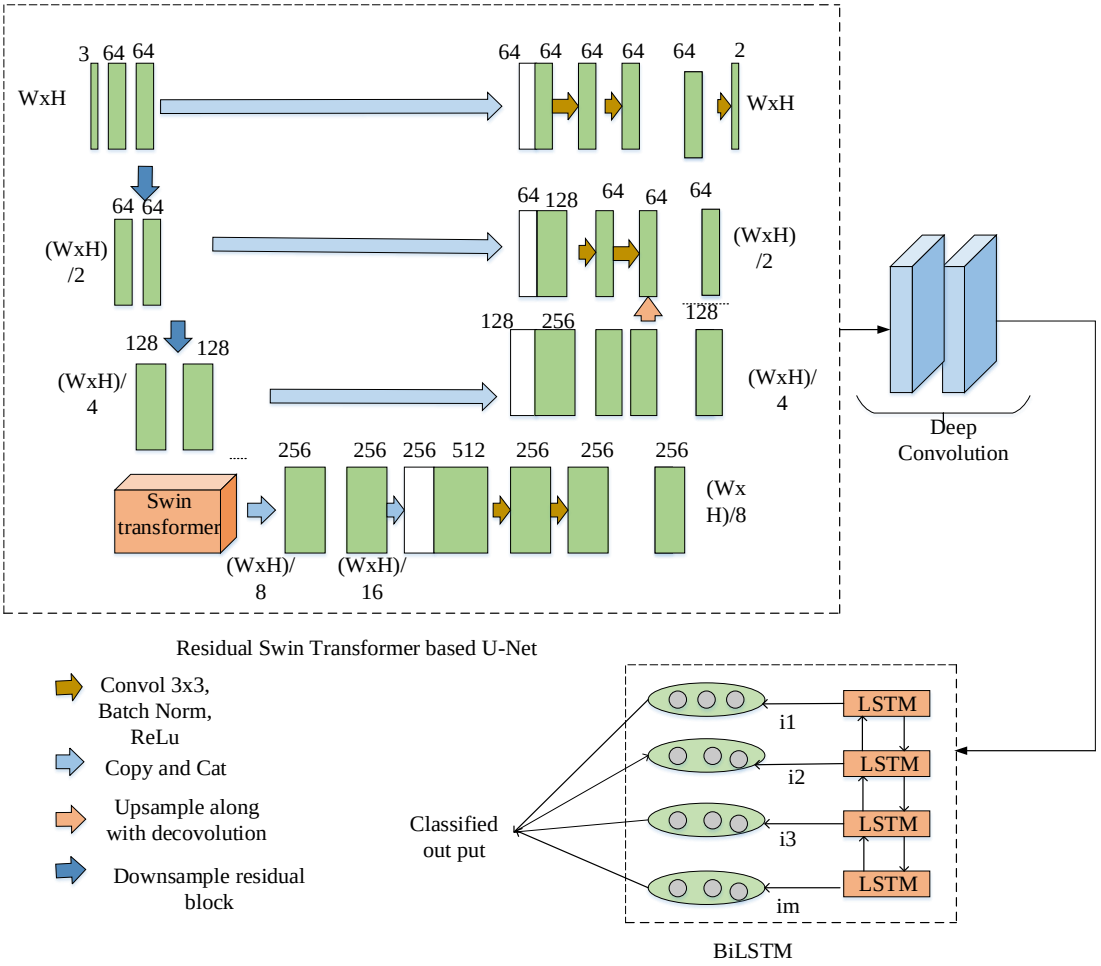


Fig. 3. Triple task learning framework

For the standard transformer network, each token wants to calculate its correlation with every other token. Here, the computational complexity is high, which is equal to the amount of tokens, thereby making it unusable for tasks with high-resolution images. Therefore, for effective modelling, a swin transformer is utilized, and it consists of multi-layer self attention (MSA) and also Shifted window-dependent MSA (SW-MSA).

In W-MSA, the input feature is split into non-overlapping windows, and every window consists of NxN patches. W-MSA will only execute self-attention on local windows in this case. $\hat{I}_p$ and I_p specifies the result of W-MSA as well as MLP in p^{th} layer and can be calculated as :

$$\hat{I}_p = W - MSA\left(LN\left(I_{p-1}\right)\right) + I_{p-1} \tag{12}$$

$$I_p = MLP\left(LN\left(\hat{I}_p\right)\right) + \hat{I}_p \tag{13}$$

Here, important information interaction is lacking in W-MSA, so to perform cross-window interaction, a shifted window MSA is placed following W-MSA. The window configuration of SW-MSA is varied from the existing W-MSA layer, where it has an effective batch-processing technique.

Decoder: In this case, the segmentation framework makes use of the decoder component of the swing transformer. Upsampling, a swin transformer, and skip connections are all included in the decoder. The input features are upsampled by two at each level of the decoder and then coupled with a suitable skip connection of the encoder's feature maps. The output is then routed to the swing transformer unit. This architecture is selected because it permits the utilization the entire feature from the upsampling as well as the encoder. It may also globally establish dependencies of long-range and decoder context interaction.

After the above stages, the output is obtained having a resolution of $\frac{H}{4}\times\frac{H}{4}$. Here, many shallow features are lost while performing $4\times$ upsampling. Hence, the input image is down-sampled by combining 2 units to obtain the low-level feature having a resolution of $\frac{H}{2}\times\frac{H}{2}$ as well as $H\times W$. Here, every block comprises of 3×3 convolution layer, a ReLU layer and a group normalization layer. By using the skip connection, all of the output features are used to anticipate the final mask. As a result, the impacted area is split using a U-net based on residual swin transformers.

3.3 Feature Extraction using Deep Convolutional Neural Network

After segmentation, the segmented image is given to module 2 for feature extraction. Here, feature extraction is carried out using a Deep convolutional neural network. Here, dense layers are added in the initial part of the architecture. Fig.4 illustrates the architecture of a deep convolutional neural network.

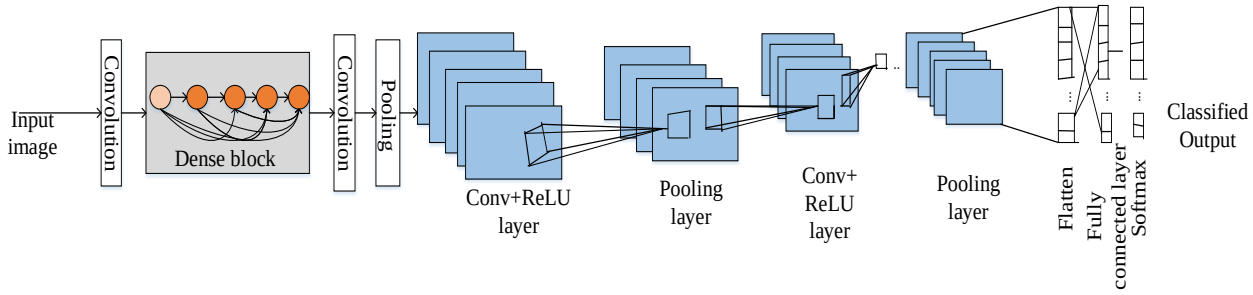


Fig. 4. Deep convolution neural network

As a feed-forward strategy, each layer in the DenseNet design was linked to each other layer. The dense layer is a significant part of DenseNet that enhances the flow of information between the layers. DenseNet comprises ReLU, batch normalization and 3x3 convolution. The mathematical formula is given by

$$a_1 = Z_k\left(\left[a_0, a_1, \ldots a_{k-1}\right]\right) \tag{14}$$

Here, $\left[a_0, a_1, \ldots a_{k-1}\right]$ specifies the integration of feature maps generated in layers 0,1,…..k-1, $H_k(\cdot)$ is represented as the composite function for 3 successive operations performed on the k^{th} layer.

In a deep convolution neural architecture, the image is considered as the input and from the image, extraction of features is done utilizing convolution, pooling layer and ReLU. Depending on the feature, the data is categorized utilizing a Softmax and fully connected layer. Here, the filter slides over the entire image and the pixel of the image is multiplied with the filter, and then addition is performed. After the convolution layer comes the non-linear activation function ReLU, which performs threshold operations. Negatives are converted to zero by the ReLU function. The pooling layer reduces the network's complexity in this network. The rectangle region is gathered here, and the highest value or average of that area is calculated. By combining a number of ReLU, convolution and pooling, the features get extracted. The rectangular region is pooled here, and the largest value is considered, or the average of that area is calculated. Depending on the probability score, the classification unit categorizes the image.

3.4 Classification using Extended Bidirectional Long Short Term Memory (EBiLSTM)

After extracting the features present in the input image, the features are fed to the classification phase to categorize the various classes of COVID-19. In this work, Extended BiLSTM is utilized for classification purposes. The recurrent neural architecture contains the limitation of huge time dependence. So, the memory cells contained in the LSTM architecture can save the image data with a large time sequence. The information is sent via a gate network having only less interactions. Here, information waste is reduced, allowing storage cells to hold critical information. The LSTM comprises three gate structures, namely, input gate i_r, output gates Z_r, and forget gates g_t. At time r, the LSTM has an input a_r, k_{r-1} and the output is k_r, b_r. Initially, at the moment r read the input and a sigmoid function is utilized to compute the loss of information by the forget gate cells.

$$g_r = \sigma\left(V_g.[k_{r-1}, a_r] + y_g\right) \tag{15}$$

The input gate was utilized to find what information is to be updated in the cell.

$$i_r = \sigma\left(V_i \cdot \left[k_{r-1}, a_r\right] + y_i\right) \tag{16}$$

Here, g_r multiplied the previous cell state N_{r-1} and adds information of the new cell for finishing the cell information update.

$$N_r = g_r \times N_{r-1} + i_r \times \tilde{N}_r \tag{17}$$

The output gate is used to determine the information about the cell state that will be output.

$$Z_r = \sigma\left(V_0 \cdot \left[k_{r-1}, a_r\right] + y_0\right) \tag{18}$$

Lastly, utilize the tanh function to compute the cell information of the output.

$$k_r = Z_r \times \tanh N_r \tag{19}$$

Here, y_g , y_n, y_i, y_0 etc. denote the bias operation, V_g, V_n, V_i, V_0 etc. denotes the weights and also σ specifies the sigmoid operation. Here, BiLSTM combines the backward and forward outputs of the LSTM at r^{th} moment.

$$g_r = \left[\overrightarrow{k_r}; \overleftarrow{k}\right] \in T \tag{20}$$

In this work, the BiLSTM acquires the input as the preceding layer's feature fusion vector and also random initialization is assigned for the weight V as well as bias function y. The LSTM block computes the forget gate value g_r, output gate z_r and input gate i_r, depending on the information present in the hidden unit and storage unit. Lastly, the output vectors of the next and previous moments of LSTM blocks are provided as the output of the BiLSTM. Thus, the BiLSTM classifies the image into various classes of COVID-19. The proposed triple-task learning framework addresses the overfitting problem by using L1-regularization method. This method can assign penalty for the higher weights and helps to avoid learning of irrelevant features. The L1 regularization allocates the weight range as zero for the unwanted features. Also, this regularization technique promotes sparsity by learning minimal features and thereby avoids complexity and overfitting issues. Therefore, the utilization of L1 regularization enhances the classification accuracy result.

3.4.1 Adaptive Fire Hawks Algorithm

The hyperparameters are fine-tuned utilizing the Adaptive Fire Hawks Algorithm. During classification, a loss function known as the cross entropy loss emerges, which is represented by the following equation (21).

$$CE = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{b} b_{ij} \log g_j\left(a_i; \theta\right) \tag{21}$$

Here, θ represents the list of classifier parameters, b_{ij} represents the j^{th} element for the one-hot encoded label of the image a_i. g denotes the empirical risk of the classifier, g_j denotes the j^{th} element of g.

In order to minimize the cross entropy loss the hyperparameters should be fine-tuned. In this work, the Adaptive Fire Hawks Algorithm is utilized for fine-tuning the hyperparameters. It is a metaheuristic algorithm in which the fire hawks follow a specific strategy to catch their prey. This strategy is utilized to optimize the hyperparameters. Here, the fire hawks are considered search agents, and their prey is considered the parameter. The algorithm operates on the following fitness function given in equation (22).

$$Fitness = \max[accuacy] \tag{22}$$

Here, the search agent updates its location using the following equation (23)

$$SA_k^{new} = SA_k + (t_1 \times gb - t_2 \times SA_{near}), k = 1,2,.....m \quad (23)$$

Here, SA_k^{new} refers to the new location vector of the k^{th} search agent. gb refers to the global best result in the search area. SA_{near} is one among other search agent in the search area. $t_1 and t_2$ represent the uniform random numbers in the interval [0,1].

For every iteration, the fitness is calculated, and until the condition is satisfied, the position of the search agent is updated to locate the optimal parameter.

4. Result Analysis and Discussion

In the result analysis section, the experimental output of the proposed triple-task learning network is explained in detail. Here, simulation is carried out using Python tool, and the proposed framework is compared with four existing techniques, namely U-Net with BiLSTM (U-Net_BiLSTM), Residual U-Net with convolutional BiLSTM (Res U-Net_CBiLSTM) and U-Net with Extended BiLSTM (U-Net_EBiLSTM). The data set utilized for experimentation is provided in the link below:

https://github.com/UCSD-AI4H/COVID-CT

The dataset contains 746 CT scans collected from 216 COVID-19 patients. The utilized dataset involves four different severity classes like no-covid, mild, medium and severe. The total data available in no-covid cases are 394, mild are 218, medium are 83 and 51 of severe cases. The proposed study have not utilized any specific criteria for labelling the severities since it is already involved pre-assigned labels in the utilized dataset such as 0, 1, 2 and 3. Whereas, 0 indicates no-covid, 1 indicates mild, 2 represents medium and 3 indicates as severe. In order to mitigate the class imbalance issue, class weighting strategy is applied, which allocates higher misclassification penalties to minority class samples like medium and severe while training the model. Thus, the proposed study avoids class imbalance issues and making it suitable for accurate classification process. For evaluation, the dataset is splitted into 80:20 for training and testing process. The proposed technique's performance is evaluated using metrics such as accuracy, specificity, precision, recall, F1 score, Root Mean Square Error (RMSE), Mean Square Error (MSE), and Mean Absolute Error (MAE). Table 1 depicts the system configuration used for experimental verification.

Table 1. System configuration

Device name	SMG116
Full device name	SMG116.smg.local
Processor	Intel®core™i5-4670 CPU @3.40 GHz 3.40 GHz
Pen and touch	No pen or touch input is available for this display
System type	64-bit operating system,x64-based processor
Installed RAM	8.00 GB (7.80 GB usable)

4.1 Performance metrics

- ***Accuracy***

It is known as the estimation for the nearness of the observed value to the reference value. It can be mathematically expressed as in the below equation.

$$\text{Accuracy} = \frac{TP + PN}{TP + TN + FP + FN} \quad (24)$$

Here, TN specifies the True Negative value, FP specifies the False Positive value, TP refers to the True Positive value, and FN gives the False Negative value.

- ***Precision***

In the case of predicted values, precision is defined as the ratio of true positive to total true positive and false positive values. It is mathematically given by

$$\text{Precision} = \frac{TP}{TP + FP} \quad (25)$$

- ***Recall***

The ratio of observed true positive discoveries to total projected true positive and false negative values is known as recall. It can be seen in the following equation (26):

$$\text{Recall} = \frac{TP}{TP + FN} \tag{26}$$

- ***F1 score***

F1 Score is called the variant of accuracy, which will not be influenced by the negatives. It can be mathematically represented as in the following equation (27).

$$2\frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{27}$$

- ***Specificity***

The ratio of true negatives to total predicted values as false positives and true negatives is referred to as specificity. It is given by

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{28}$$

- ***Mean Square Error (MSE)***

MSE is referred to as the measure of the mean squared difference among the predicted values as well as actual values. If no error is present, the value of MSE is Zero. It is demonstrated in the following equation (29).

$$MSE = \frac{1}{m}\sum_{i=1}^{m}\left(P_i - \hat{P}_i\right)^2 \tag{29}$$

Here, P_i represents the observed values, $\hat{P}_i$ denotes the attained values and m denotes the total amount of observations.

- ***Root Mean Square Error (RMSE)***

RMSE is defined as the square root of the mean square difference between predicted and observed values. It is expressed in the following equation (30).

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(P_i - \hat{P}_i\right)^2} \tag{30}$$

- ***Mean Absolute Error (MAE)***

It is referred to as the average total of absolute errors. It is expressed mathematically in the following equation (31).

$$MAE = \sum_{i=1}^{m}\left|P_i - \hat{P}_i\right| \tag{31}$$

4.2 Performance analysis and comparison

The performance measures are used to assess the outcomes of the proposed technique. Furthermore, the proposed approach is compared to three existing procedures, and the comparison findings show that the proposed approach outperforms the current ways.

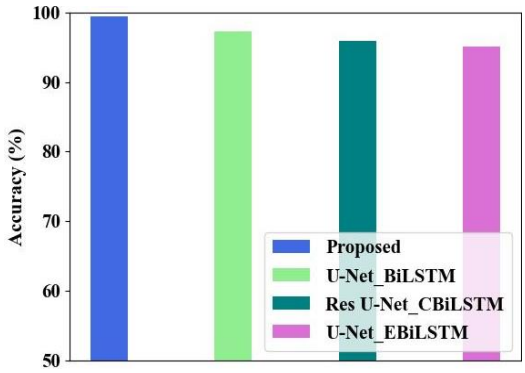


Fig. 5. Accuracy Comparison

Fig.5 depicts the comparison of the proposed technique's accuracy with three existing techniques, namely U-Net_BiLSTM, Res U-Net_CBiLSTM and U-Net_EBiLSTM. Comparison output illustrates the proposed approach is superior to existing techniques. According to the graph, the proposed study obtains an accuracy of 99.46%, which is higher than the existing approaches. But for the existing mechanisms, U-Net_BiLSTM shows an accuracy of 97.18%, Res U-Net_CBiLSTM has an accuracy of 95.84%, and U-Net_EBiLSTM achieves an accuracy of 95.04%.

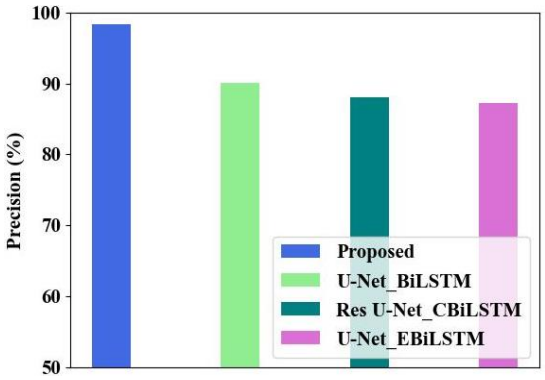


Fig. 6. Precision Comparison

Fig.6 illustrates the precision comparison of the proposed approach to other existing approaches like U-Net_BiLSTM, Res U-Net_CBiLSTM and U-Net_EBiLSTM. The graph clearly shows that the proposed study attains a better precision value than the existing techniques. The proposed technique achieves a precision of 98.31%, which is better than that of existing techniques. In the case of U-Net_BiLSTM, it achieves a precision of 90.11%, Res U-Net_CBiLSTM attains a precision of 88.03%, and U-Net_EBiLSTM has a precision value of 87.29%, and it is lower than the proposed technique.

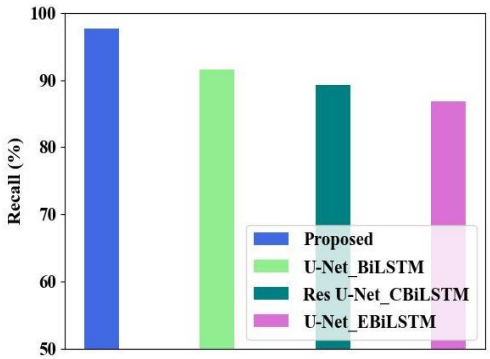


Fig. 7. Recall Comparison

Recall comparison of the proposed approach to other existing approaches such as U-Net_BiLSTM, Res U-Net_CBiLSTM and U-Net_EBiLSTM is depicted in Fig.7. The proposed technique achieves a recall of 97.63%, which is higher compared to existing techniques. But for the existing techniques, U-Net_BiLSTM achieves a recall of 91.52%, Res U-Net_CBiLSTM attains a recall of 89.28%, and U-Net_EBiLSTM has a recall value of 86.85% and this lower than the proposed technique.

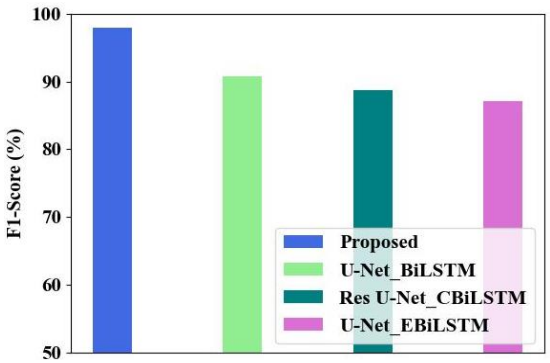


Fig. 8. F1 score comparison

Fig.8 illustrates the F1 score comparison of the proposed approach to other existing approaches such as U-Net_BiLSTM, Res U-Net_CBiLSTM and U-Net_EBiLSTM. The graph reveals that the proposed technique obtained a good F1 score, which is greater than that of the existing approaches. The proposed technique achieves an F1 score of 97.97%, which is higher than that of the existing techniques. But for the existing techniques, U-Net_BiLSTM achieves an F1 score of 90.81%, Res U-Net_CBiLSTM attains an F1 score of 88.65%, and U-Net_EBiLSTM has an F1 score of 87.07%, and this is lower than the proposed technique.

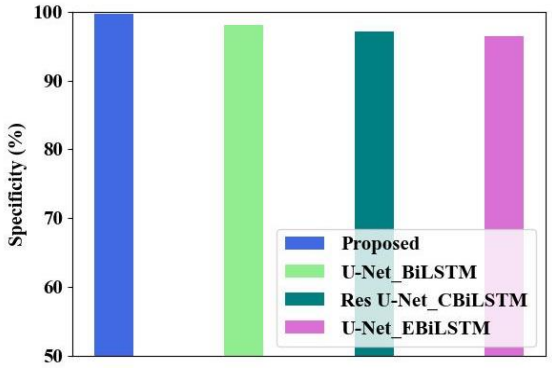


Fig. 9. Comparison of specificity

Fig.9 illustrates the Specificity comparison of the proposed approach with other existing approaches such as U-Net_BiLSTM, Res U-Net_CBiLSTM and U-Net_EBiLSTM. The graph shows that the proposed approach has a high Specificity when compared to existing approaches. The proposed approach achieves a specificity of 99.65%, which is higher when compared to the existing techniques. But for the existing techniques, U-Net_BiLSTM achieves a specificity of 98.08%, Res U-Net_CBiLSTM attains a specificity of 97.04%, and U-Net_EBiLSTM has a specificity of 96.41, and this is lower than the proposed technique.

MSE of the proposed approach is compared to other existing mechanisms, namely U-Net_BiLSTM, Res U-Net_CBiLSTM and U-Net_EBiLSTM and is demonstrated in Fig.10. It is inferred from the graph that the proposed technique attains a minimum MSE value higher than the existing techniques. The proposed technique has an MSE value of 0.014, which is lower than that of the existing techniques. But for the existing techniques, U-Net_BiLSTM incurs an MSE of 0.076 Res U-Net_CBiLSTM attains an MSE of 0.115, and U-Net_EBiLSTM has an MSE of 0.103, and this is lower than the proposed technique. This explains that the proposed approach has a minimum MSE value than the existing techniques. A system with low error values shows improved performance.

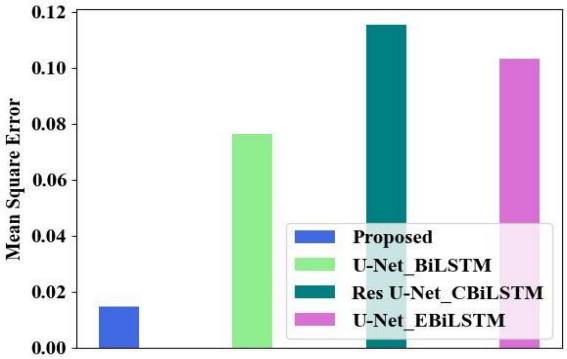


Fig. 10. MSE comparison

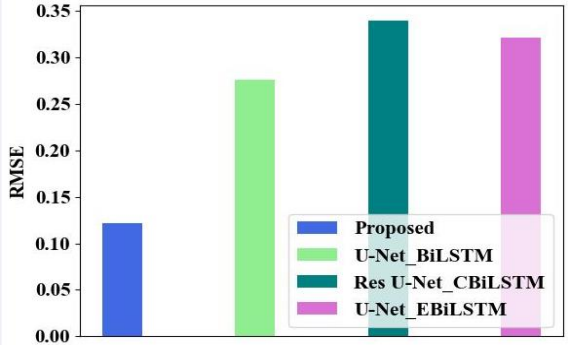


Fig. 11. RMSE Comparison

RMSE comparison of the proposed approach to other existing approaches, namely U-Net_BiLSTM, Res U-Net_CBiLSTM and U-Net_EBiLSTM, is shown in Fig.11. It is inferred from the graph that the proposed technique achieves a lower RMSE value than the existing techniques. The proposed technique has an RMSE value of 0.121, which is lower than that of the existing techniques. But for the existing techniques, U-Net_BiLSTM incurs an MSE of 0.28, Res U-Net_CBiLSTM incurs an RMSE of 0.34, and U-Net_EBiLSTM shows an RMSE of 0.32, and this is lower than the proposed technique. A system with low error values shows improved performance.

MAE comparison of the proposed approach is illustrated in Fig.12. The proposed technique achieves a lower MAE value than the existing techniques. The proposed technique has an MAE value of 0.012, which is lower than that of the existing techniques. But for the existing techniques, U-Net_BiLSTM incurs an MAE value of 0.063, Res U-Net_CBiLSTM has an MAE of 0.093, and U-Net_EBiLSTM shows an MAE of 0.1, and this is lower than the proposed technique. A system with low error values shows improved performance.

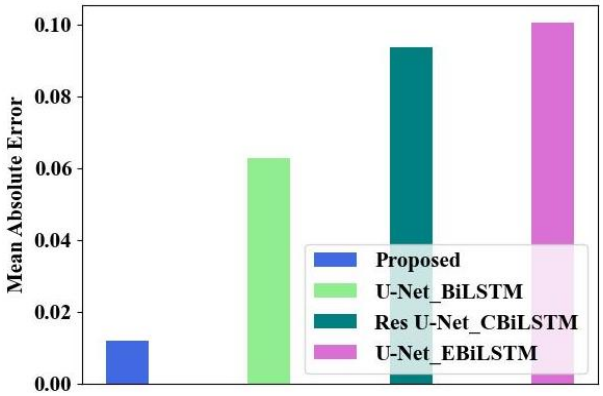
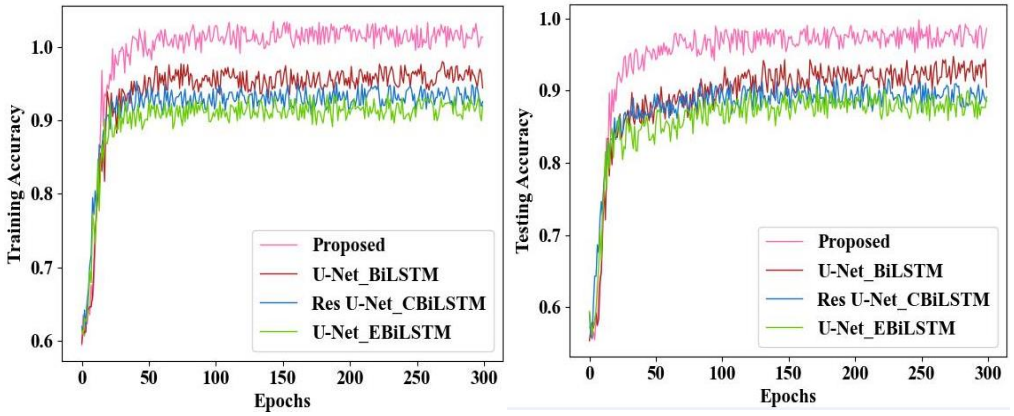


Fig. 12. MAE Comparison

Table 2. Performance metrics Comparison

Performance metrics	Techniques			
	U-Net_BiLSTM	Res U-Net_CBiLSTM	U-Net_EBiLSTM	Proposed
Accuracy	97.18	95.84	95.04	**99.46**
Precision	90.11	88.03	87.29	**98.31**
Recall	91.52	89.28	86.85	**97.63**
Specificity	98.08	97.04	96.41	**99.65**
F1-score	90.81	88.65	87.07	**97.97**
MSE	0.076	0.115	0.103	**0.014**
RMSE	0.28	0.34	0.32	**0.121**
MAE	0.063	0.093	0.1	**0.012**

Table 2 shows the performance metrics comparison of the proposed as well as the existing techniques. The table reveals the performance effectiveness of the proposed mechanism.



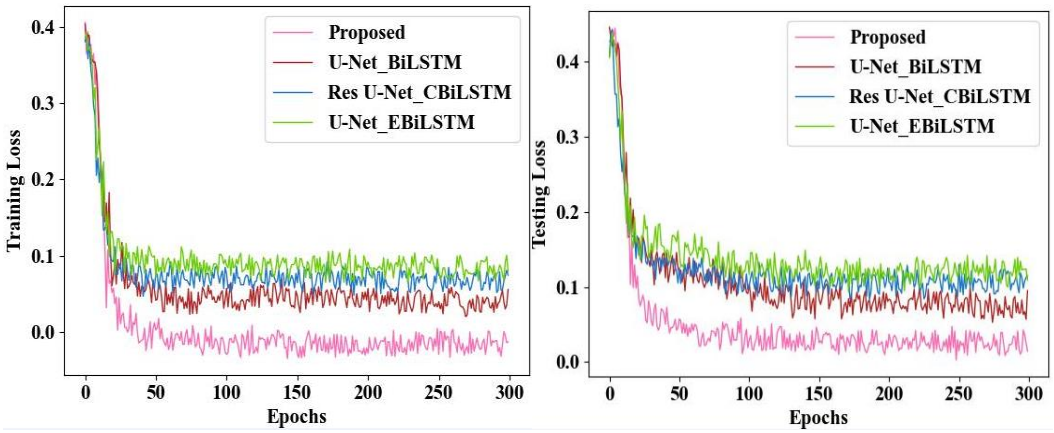
(a) Training Accuracy (b) Testing Accuracy

Fig. 13. (a, b) Training and Testing accuracy curves

Fig.13 (a) reveals the training accuracy curves of the proposed and existing approaches. Here, when the epoch is zero, the training accuracy starts increasing and reaches a peak when the epoch is 50. After that, the training accuracy remains constant throughout the training period. The testing accuracy curves of proposed and existing techniques are

given in Fig.13 (b). Here, when the epoch is zero, the testing accuracy starts increasing, and then it gradually increases up to the 100th epoch. Then, the testing accuracy remains constant throughout the testing period.

Fig.14 (a) illustrates the training and also testing loss curves of the proposed and existing techniques. Here, when the epoch is zero, the training loss starts declining and reaches a minimum when the epoch is 50. After that, the training loss remains steady during the entire training process. Fig.4 (b) illustrates the testing loss curves of proposed and existing techniques. Here, the testing loss decreases upto 50th epoch and then remains constant throughout the testing period.



(a) Training loss (b) Testing loss

Fig. 14. (a, b): Training and testing loss curves

Fig.15 illustrates the confusion matrix for the proposed technique. Here, the confusion matrix proves the effectiveness of the proposed technique to classify various classes of COVID-19. Here, out of 394 non-COVID samples tested, 393 of them are correctly classified as No Covid, and only one sample is misclassified as a mild class. Also, out of 218 mild samples tested, 217 are correctly classified as a mild class, and only one is wrongly classified as a medium class. Moreover, out of 83 medium-class samples tested, 79 are correctly classified as a medium class and only one sample is misclassified as a severe class, 3 are misclassified as a mild class. Then, out of 51 severe class samples tested, 49 were correctly classified as severe class, and one was misclassified as medium and severe class.

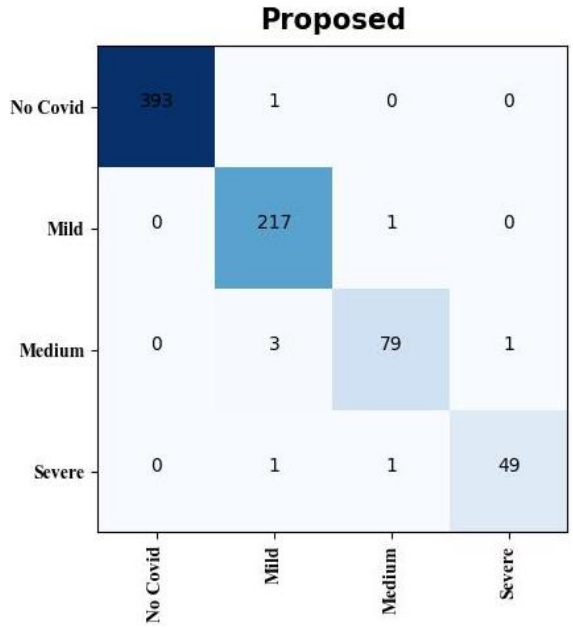


Fig. 15. Confusion matrix

Fig.16 depicts the accuracy comparison of proposed and existing strategies by altering the learning rate. The proposed technique has a 91% accuracy when the learning rate is 0.05. At a learning rate of 0.01 the proposed technique has a 93.2% accuracy. For a learning rate of 0.005, the proposed approach achieves an accuracy of 96.26%, which is much superior to existing techniques. The proposed framework has an accuracy of 99.46% at a learning rate of 0.001, which is higher than the prior methodologies. It is clear from the graph that as the learning rate improves, so will the accuracy. Table 3 shows the accuracy comparison in terms of learning rate.

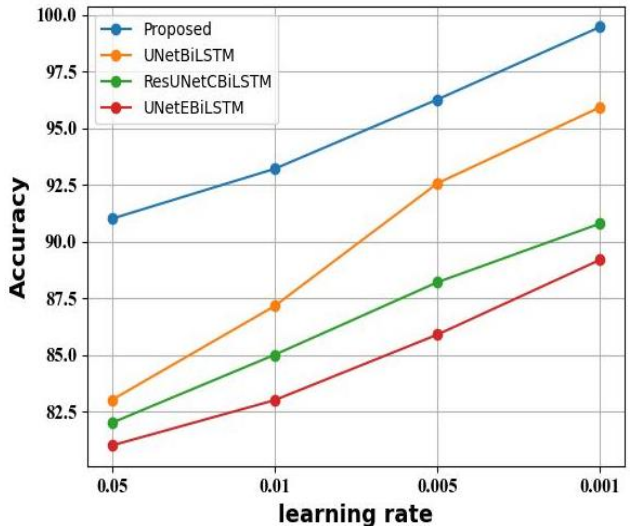


Fig. 16. Accuracy with respect to learning rate

Table 3. Accuray in terms of learning rate

Learning rate	Accuracy (%)			
	Proposed	U-Net_BiLSTM	Res U-Net_CBiLSTM	U-Net_EBiLSTM
0.05	**91**	83.01	82	81
0.01	**93.2**	87.15	85	83
0.005	**96.26**	92.56	88.2	85.89
0.001	**99.46**	95.92	90.79	89.19

Fig.17 shows the accuracy comparison of proposed and existing techniques by varying the batch size. When the batch size is 16, the proposed technique achieves an accuracy of 99.46%, but for the existing techniques, the accuracy is lower. For a batch size of 32, the proposed technique has 96% accuracy, which is higher than that of the existing techniques. If the batch size is 64, it achieves 92.5% accuracy, which is better. For a batch size of 128, the accuracy of the proposed technique is 90.4%, which is higher than that of the existing techniques. This shows the efficacy of the system proposed over other existing techniques.

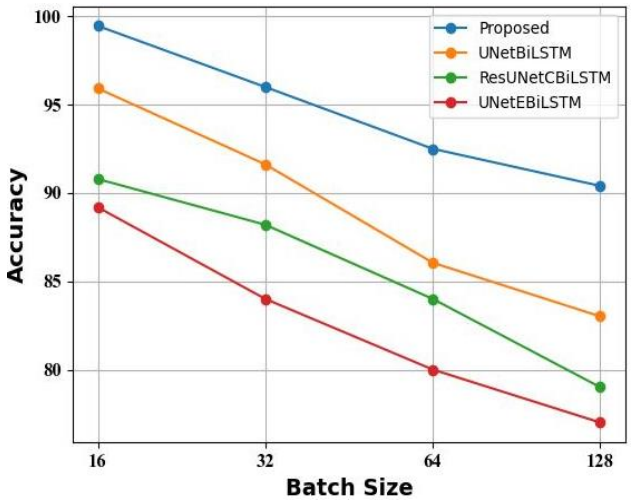


Fig. 17. Accuracy in terms of batch size

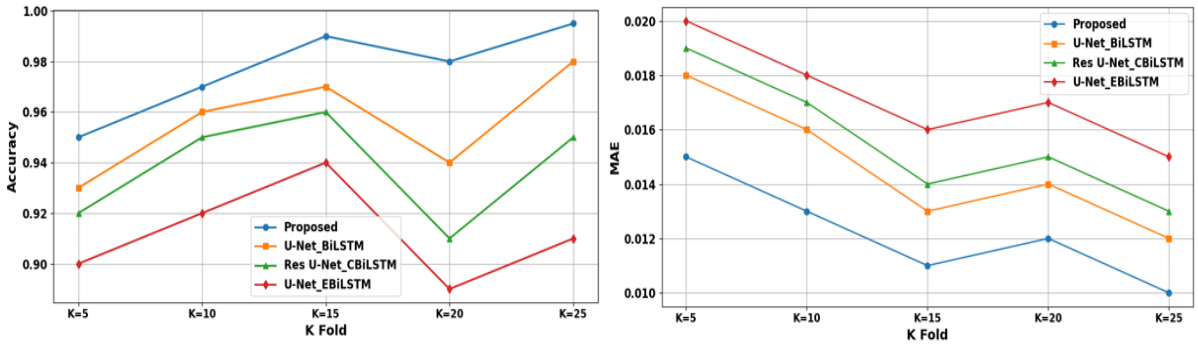
Table 4. Accuracy in terms of batch size

Learning rate	Accuracy (%)			
	Proposed	**U-Net_BiLSTM**	**Res U-Net_CBiLSTM**	**U-Net_EBiLSTM**
16	**99.46**	95.92	90.79	89.19
32	**96**	91.62	88.2	84
64	**92.5**	86.05	84	80
128	**90.4**	83.01	79	77

Table 4 demonstrates the accuracy of the proposed as well as existing techniques in terms of batch size. The graph clearly shows that as the batch size increases, the accuracy decreases. Thus, the overall results show that the triple-task learning framework proposed in this work is better than the existing techniques.

4.3. Empirical validation using K-fold cross validation

In this section, the performance of proposed classification model is further validated through K-fold cross validation. In this accuracy and MAE of proposed classifier is evaluated and the results are compared with other existing methods like U-Net-BiLSTM, Res U-Net_CBiLSTM and U-Net-EBiLSTM. In this, the K-values are varied from 5 to 25 and analyze the efficacy of proposed model. Fig. 18 (a)-(b) illustrates the accuracy and MAE comparison in terms of K-fold cross validation.



(a) Accuracy comparison (b) MAE comparison

Fig. 18. (a)-(b): Accuracy and MAE in terms of K-fold validation

The comparison analysis clearly proves that the proposed model is superior to other existing methods. By varying the K-folds from 5 to 25, classification performance of proposed model is enhanced. Similarly, the misclassification error is reduced and is clearly represented in MAE analysis. As compared with existing methods, the proposed triple task learning model is highly efficient. Because of the incorporation of swin transformer U-net, spatial information and long-term dependencies are easily gets captured. Thus, the affected portions are accurately segmented from the given samples and helps the classification model to make precise decision. Also, the combination of each components in the triple-task learning framework contributes the classification performance.

4.4 Ablation study analysis

This analysis exhibits the individual contribution of each components that proposed in this work. Thorough this evaluation, the significance of each components in classification performance is analyzed. Here, the analysis is performed by five different modules. Module 1 shows the classification accuracy of proposed model without the presence of pre-processing step. Module 2 indicates the accuracy result of proposed classification by neglecting the swin transformer block in the segmentation stage. Module 3 shows the results without utilizing BiLSTM in the triple-task learning framework. Module 4 shows the accuracy results by skipping the fine-tuning process in the final step and module 5 shows the accuracy result of proposed model including all components. Table 5 shows the ablation study evaluation of proposed work.

Table 5. Ablation study analysis

Modules	**Accuracy (%)**
Module 1 (Without pre-processing)	92.23
Module 2 (Without swin transformer)	87.56
Module 3 (Without BiLSTM)	95.34
Module 4 (Without Fire Hawk Algorithm)	96.45
Proposed (Including all components)	99.46

The analysis shown in Table 5 clearly states the necessity of each components that used in the proposed triple task learning framework. By neglecting the pre-processing step, the model produce reduced accuracy result as 92.23% and it clearly exhibits the need of pre-processing using Gabor filtering and image re-sizing. Similarly, while excluding the swin transformer component, the proposed model produce the accuracy of 87.56% in module 2. As compared with all other modules, module 2 shows the lowest accuracy rate. This is because, by neglecting the swin transformer block, classification accuracy is reduced because of the inefficient segmentation process. The utilization of swin transformer in the segmentation stage helps to capture hierarchical feature representation and learns the complex features efficiently. On the other hand, the classification accuracy rate is reduced as 95.34% without using BiLSTM block. BiLSTM has the ability to fetch the features in both forward and backward directions and attains contextual dependencies from the feature maps. Also, the accuracy is reduced as 96.45 because of skipping the fine-tuning process using Fire Hawk Algorithm. The fine-tuning model helps to reduce the parameters and improves the learning ability. By incorporating all the components, the proposed study have attained an improved accuracy rate of 99.46%. Thus, the analysis proves the importance of each components that proposed in this work for robust classification task.

4.5 Performance evaluation under different train-test splits

The proposed study have evaluated the results by splitting the dataset into 80:20 for training and testing. In this case, the experimental results are also further validated by explicitly describing the train-test split ratio under 70:30 and 60:40. When the data distribution is higher in training set as compared with testing, it can cause overfitting problem. Also, the larger testing set can cause learning inabilities. But, the proposed model performs consistently under varied split ratios and proves the efficacy of proposed triple-task learning framework. Fig. 19 shows accuracy comparison analysis under 80:20, 70:30 and 60:40 splitting ratios.

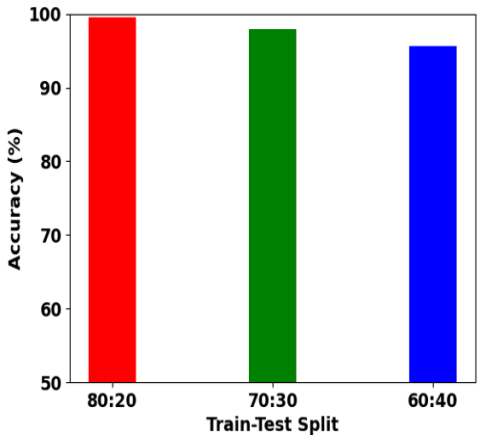
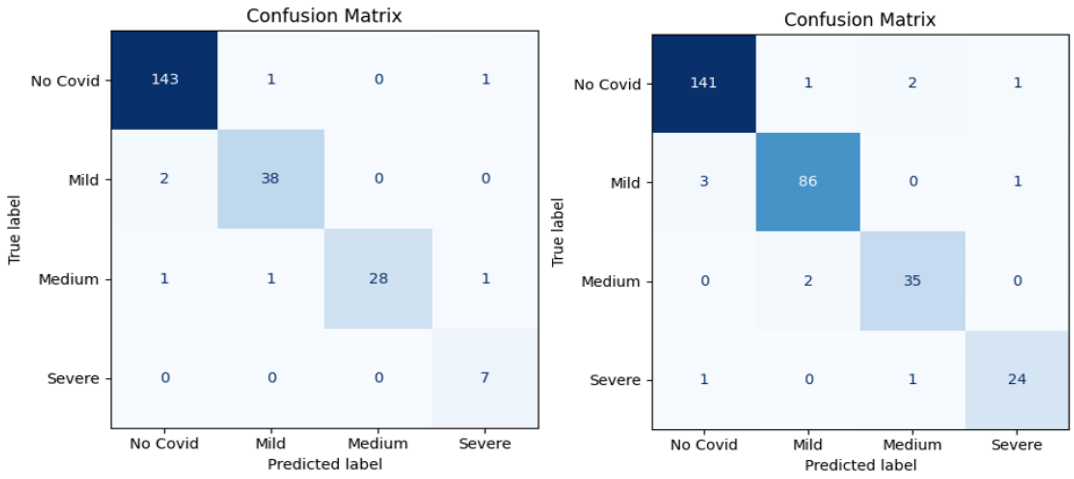


Fig. 19. Accuracy comparison under different split ratios

The analysis under different training and testing split ratios clearly proves the efficiency of proposed work. As compared with other classification models, the proposed model obtains better accuracy rate in 80:20, 70:30 and 60:40 splitting ratios. Thus, the analysis proves the generalizability of proposed model and validates its strength in COVID-19 classification task. Also, for evaluating the misclassification errors under different train-test split ratios, confusion matrix is plotted and illustrated in Fig. 20 (a)-(b).



(a) Analysis under 70:30 ratio (b) Analysis under 60:40 ratio

Fig. 20. (a)-(b): Confusion matrix analysis under 70:30 and 60:40 train-test split ratios

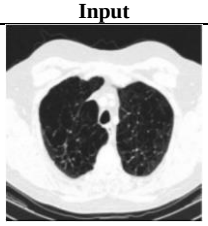
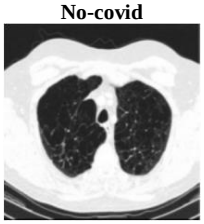
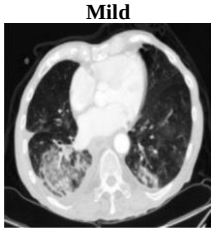
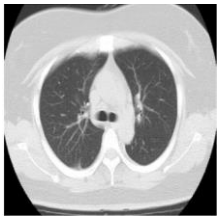
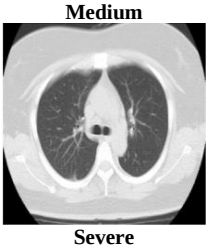
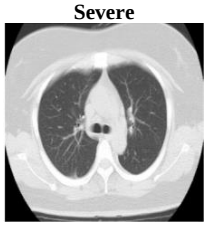
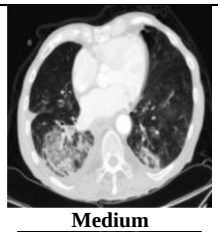
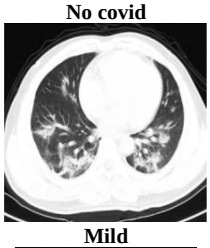
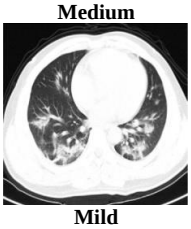
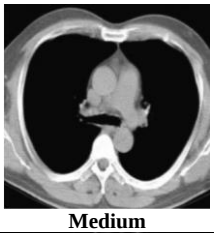
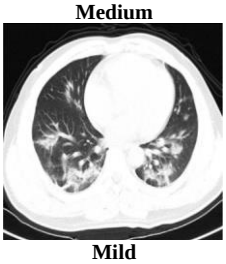
Dataset train-test split ratios	Input	Classified outcomes
70:30	 No-covid	 No-covid
	 Mild	 No-covid
	 Medium	 Severe
	 Severe	 Severe
60:40	 No covid	 Medium
	 Mild	 Mild
	 Medium	 Medium
	 Severe	 Mild

Fig. 21. Sample input and classified images

In Figure 20 (a) the confusion matrix of proposed model is provided and it shows the misclassification rate of proposed model in COVID-19 classification. From the total of 145 no-covid cases, only 2 cases are misclassified as mild and severe. Similarly, from the total of 40 mild cases, only 2 cases are wrongly classified as no-covid and remaining 38 are correctly classified. On the other hand, from the total of 31 medium cases, only 3 cases are wrongly classified. Finally, by testing with 7 severe cases, all cases are correctly classified as severe. Thus, by splitting the dataset as 70:30 for training and testing, the proposed model provides better classification performance. In addition, while the dataset is splitted into 60:40, the classification efficacy of the proposed model is enhanced as shown in Figure 20 (b). In this, from the total count of 145 no-covid cases, 141 are correctly classified and only 4 cases are misclassified. Also, the mild severity involves 90 cases in which 86 are correctly classified as mild and only 4 are misclassified. Similarly, the medium severity contains 37 cases, where only 2 cases are wrongly classified. Finally, from the total of 26 severe cases, 24 cases are correctly classified and only two cases are misclassified. Thus, the analysis clearly shows the effectiveness of proposed model however fewer misclassifications are occurred due to the less interpretability. It can be addressed in the future by utilizing Explainable Artificial Intelligence (EAI) methods and further improves the classification efficiency. Some visual examples of correctly and incorrectly classified cases are shown in Fig. 21.

4.6 Computational complexity analysis

The complexity of proposed model is analyzed to further validate the robustness of proposed triple-task learning framework. In this, training time, model complexity and inference speed of proposed model is evaluated. For classifying the COVID-19 disease, the proposed model requires the overall training time of 32.5 hrs for 300 epochs, 11.21 hrs for 100 epochs and inference speed of 0.51sec. Also, the model's complexity is analyzed through Big 'O' notation which portrays the time and space complexities and compared the efficiency with other existing methods. Table 6 shows the computational complexity comparison analysis.

Table 6. Computational complexity comparison

Methods	Complexity
Proposed (Triple task learning framework)	$O(1)$
U-Net-BiLSTM	$O(\log_2 N)+O(1)$
Res U-Net_CBiLSTM	$O(nr2\log r)+O(nT)$
U-Net_EBiLSTM	$O(\log_2 N)+O(1)+O(nT)$

Thus, the results shown in complexity analysis clearly illustrates the effectiveness of proposed classification model. As compared with other existing models, the proposed study have attained reduced training time, improved inference speed and reduced computational complexity. Because of using efficient components in the proposed triple-task learning framework, the complexity issues are mitigated without impacting the classification performance.

4.7 Experimental analysis using other dataset

For analyzing the generalizability of proposed model, different dataset is considered and performs the classification process. For this, the dataset is collected from different sources which includes Chest X-Ray and CT scan images [26]. The utilized dataset contains totally 1480 samples in which 676 images are covid-19 cases and remaining 804 images are non-covid-19. The dataset is distributed into 1008 for training and 472 for testing. By using this dataset, the proposed triple-task learning framework classifies the samples and provides better outcomes than other compared methods like UNAS-Net, InceptionNet, VGG-16 and ResNet-50. Table 7 shows the comparison analysis under different dataset.

Table 7. Performance comparison under different dataset [26]

Models	Accuracy (%)	AUC
UNAS-Net	97.36	98.08
InceptionNet	98.18	98.05
ResNet-50	85.45	92.63
VGG-16	98.18	99.97
Proposed	**99.51**	**99.97**

Thus, the attained results shows that the proposed model has the ability to handle different dataset because of achieving improved accuracy rate and it proves the generalizability of the proposed model.

4.8 Comparison analysis with recent existing state-of-the-art methods

The proposed work is compared with several state-of-the-art methods for showing the efficacy of proposed triple-task learning framework. Table 8 shows the detailed comparison analysis.

Table 8. Comparison analysis with recent state-of-the-art methods

Authors	Methods	Accuracy (%)
Ruffini et al. [27]	Multi-dataset multi-task training framework	68.6
Sharma et al. [28]	Multi-task attention network	91.84
Sharma et al. [28]	Multi scale attention based DenseNet-201	96.71
Kordnoori et al. [29]	Deep multi-task learning structure	96
Kordnoori et al. [30]	Deep multi-task learning model (UNet, encoder decoder and multilayer perceptron)	95.62
Proposed	Triple-task learning framework	99.46

As compared with other existing multi-task learning framework, the proposed model has achieved improved accuracy result. This is because, the proposed model integrates different stages including segmentation, feature extraction and classification. The model incorporates Residual Swin Transformer U-Net model as the initial component, which helps to segment the affected portions effectively by capturing the relevant feature information. Also, deep CNN is used in the feature extraction process where the features are learned in hierarchical manner and the final EBiLSTM makes the classification process by fetching long term dependencies. However, the existing multi-task learning framework have faced several issues like limited and imbalanced data, lack of generalization, complexity issues, etc. But, the proposed work have addressed all these issues and achieved improved accuracy outcomes.

5. Conclusion and Future Scope

In this paper, a novel triple-task learning framework is introduced to detect and classify the COVID-19 disease. Here, pre-processing of input images is carried out using Gabor filtering and image resizing, and thus, the unwanted noise present in the image is removed effectively. After that, using the triple-task learning framework, segmentation, feature extraction and classification are carried out effectively. Here, the affected regions are segmented utilizing the Residual Swin Transformer U-Net. Then, utilizing the deep convolution network, the most important features are extracted, and thus classification accuracy is improved, thereby reducing computational complexity. After that, using the Extended BiLSTM, the various classes of COVID-19, namely, severe, mild and medium, are classified effectively. Then, in order to fine-tune the hyperparameters, the Adaptive Fire Hawks algorithm is utilized. The proposed technique is experimentally verified using the Python tool. Next, the output of the proposed approach is compared to three existing techniques, namely UNetBiLSTM, ResUNetBiLSTM and UNetEBiLSTM. Comparison results show that the proposed technique shows better performance, achieving an accuracy of 99.46%, precision of 98.31%, recall of 97.63%, F1 score of 97.97%, MAE of 0.012, MSE of 0.014, RMSE of 0.121. In the future, the proposed technique can be extended to work with large datasets that have many CT images. A graphical user interface application can also be used to help physicians during the diagnosing procedure.

References

[1] M. H. Alsharif, Y. H. Alsharif, K. O. M. Yahya, O. A. Alomari, M. A. Albreem, & A. Jahid. “Deep learning applications to combat the dissemination of COVID-19 disease: A review,” European Review for medical and pharmacological sciences, (2020).

[2] M. R. H. Mondal, S. Bharati, & P. Podder. “Diagnosis of COVID-19 using machine learning and deep learning: a review,” Current Medical Imaging Reviews, vol. 17, no. 12, pp. 1403-1418, 2021.

[3] M. M. Islam, F. Karray, R. Alhajj, & J. Zeng. “A review on deep learning techniques for the diagnosis of novel coronavirus (COVID-19),” Ieee Access, vol. 9, pp. 30551-30572, 2021.

[4] A. Shoeibi, M. Khodatars, M. Jafari, N. Ghassemi, D. Sadeghi, P. Moridian, & J. M. Gorriz. “Automated detection and forecasting of covid-19 using deep learning techniques: A review,” Neurocomputing, vol. 577, pp. 127317, 2024.

[5] I. Ozsahin, B. Sekeroglu, M. S. Musa, M. T. Mustapha, & D. Uzun Ozsahin. “Review on Diagnosis of COVID‐19 from Chest CT Images Using Artificial Intelligence,” Computational and Mathematical Methods in Medicine, vol. 2020, no. 1, pp. 9756518, 2020.

[6] Y. H. Bhosale, & K. S. Patnaik. “Application of deep learning techniques in diagnosis of covid-19 (coronavirus): a systematic review,” Neural processing letters, vol. 55, no. 3, pp. 3551-3603, 2023.

[7] M. Y. Kamil. “A deep learning framework to detect Covid-19 disease via chest X-ray and CT scan images,” International Journal of Electrical & Computer Engineering (2088-8708), vol. 11, no. 1, 2021.

[8] M. Kavitha, T. Jayasankar, P. M. Venkatesh, G. Mani, C. Bharatiraja, & B. Twala. “COVID-19 disease diagnosis using smart deep learning techniques,” Journal of Applied Science and Engineering, vol. 24, no. 3, pp. 271-277, 2021.

[9] M. Yildirim, & A. Cinar. “A Deep Learning Based Hybrid Approach for COVID-19 Disease Detections,” Traitement du Signal, vol. 37, no. 3, pp. 461-468, 2020.

[10] S. Wang, B. Kang, J. Ma, X. Zeng, M. Xiao, J. Guo, & B. Xu. “A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19),” European radiology, vol. 31, pp. 6096-6104, 2021.

[11] M. Barstugan, U. Ozkaya, & S. Ozturk. “Coronavirus (covid-19) classification using ct images by machine learning methods,” arXiv preprint arXiv:2003.09424, 2020.

[12] S. H. Kassania, P. H. Kassanib, M. J. Wesolowskic, K. A. Schneidera, & R. Detersa. "Automatic detection of coronavirus disease (COVID-19) in X-ray and CT images: a machine learning based approach," Biocybernetics and biomedical engineering, vol. 41, no. 3, pp. 867-879, 2021.
[13] M. Turkoglu. "COVID-19 detection system using chest CT images and multiple kernels-extreme learning machine based on deep neural network," Irbm, vol. 42, no. 4, pp. 207-214, 2021.
[14] H. S. Maghdid, A. T. Asaad, K. Z. Ghafoor, A. S. Sadiq, S. Mirjalili, & M. K. Khan. "Diagnosing COVID-19 pneumonia from X-ray and CT images using deep learning and transfer learning algorithms," In Multimodal image exploitation and learning 2021, Vol. 11734, pp. 99-110, 2021. SPIE.
[15] J. I. Z. Chen. "Design of accurate classification of COVID-19 disease in X-ray images using deep learning approach," Journal of ISMAC, vol. 3, no. 02, pp. 132-148, 2021.
[16] A. Narin, C. Kaya, & Z. Pamuk. "Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks," Pattern Analysis and Applications, vol. 24, pp. 1207-1220, 2021.
[17] 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," Medical image analysis, vol. 65, pp. 101794, 2020.
[18] A. Bhattacharyya, D. Bhaik, S. Kumar, P. Thakur, R. Sharma, & R. B. Pachori. "A deep learning based approach for automatic detection of COVID-19 cases using chest X-ray images," Biomedical Signal Processing and Control, vol. 71, pp. 103182, 2022.
[19] L. Jingxin, Z. Mengchao, L. Yuchen, C. Jinglei, Z. Yutong, Z. Zhong, & Z. Lihui. "COVID-19 lesion detection and segmentation–A deep learning method," Methods, vol. 202, pp. 62-69, 2022.
[20] N. S. Punn, & S. Agarwal. "Chs-net: A deep learning approach for hierarchical segmentation of covid-19 via ct images," Neural Processing Letters, vol. 54, no. 5, pp. 3771-3792, 2022.
[21] N. S. Punn, & S. Agarwal. "Chs-net: A deep learning approach for hierarchical segmentation of covid-19 via ct images," Neural Processing Letters, vol. 54, no. 5, pp. 3771-3792, 2022.
[22] G. Dhiman, V. Chang, K. Kant Singh, & A. Shankar. "Adopt: automatic deep learning and optimization-based approach for detection of novel coronavirus covid-19 disease using x-ray images," Journal of biomolecular structure and dynamics, vol. 40, no. 13, pp. 5836-5847, 2022.
[23] I. Chouat, A. Echtioui, R. Khemakhem, W. Zouch, M. Ghorbel, & A. B. Hamida. "COVID-19 detection in CT and CXR images using deep learning models," Biogerontology, vol. 23, no. 1, pp. 65-84, 2022.
[24] F. Demir, K. Demir, & A. Şengür. "DeepCov19Net: automated COVID-19 disease detection with a robust and effective technique deep learning approach," New Generation Computing, vol. 40, no. 4, pp. 1053-1075, 2022.
[25] M. Afif, R. Ayachi, Y. Said, & M. Atri. "Deep learning-based technique for lesions segmentation in CT scan images for COVID-19 prediction," Multimedia Tools and Applications, vol. 82, no. 17, pp. 26885-26899, 2023.
[26] A. Syarif, N. Azman, V. V. R. Repi, E. Sinaga, & M. Asvial. "UNAS-Net: A deep convolutional neural network for predicting Covid-19 severity," Informatics in Medicine Unlocked, vol. 28, pp. 100842, 2022.
[27] F. Ruffini, L. Tronchin, Z. Wu, W. Chen, P. Soda, L. Shen, & V. Guarrasi. "Multi-Dataset Multi-Task Learning for COVID-19 Prognosis," In International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 251-261, 2024. Cham: Springer Nature Switzerland.
[28] A. Sharma, & P. K. Mishra. "Covid-MANet: Multi-task attention network for explainable diagnosis and severity assessment of COVID-19 from CXR images," Pattern Recognition, vol. 131, pp. 108826, 2022.
[29] S. Kordnoori, M. Sabeti, H. Mostafaei, & S. S. A. Banihashemi. "An efficient deep multi‐task learning structure for covid‐19 disease," IET Image Processing, vol. 17, no. 13, pp. 3728-3745, 2023.
[30] S. Kordnoori, M. Sabeti, H. Mostafaei, & S. S. A. Banihashemi. "LungXpertAI: A deep multi-task learning model for chest CT scan analysis and COVID-19 detection," Biomedical Signal Processing and Control, vol. 99, pp. 106866, 2025.

Authors' Profiles

Krishna Bhimaavarapu holds an M.Tech in Computer Science and Engineering from GIET (JNTUH), Rajahmundry, an MBA in Human Resources and Finance from VIET (JNTUK), Visakhapatnam, and B.Tech in Computer Science and Information Technology from GMRIT (JNTUH), Rajam, Andhra Pradesh, India. With extensive academic experience, he has served in various autonomous institutions as Assistant Professor, Associate Professor and Head of the Department. He is presently serving as an Assistant Professor in the Department of Computer Science and Engineering at KL University (Koneru Lakshmaiah Education Foundation), Andhra Pradesh, India. His research interests include the Internet of Things (IoT), Machine Learning(ML), and Deep Learning(DL). He is a lifetime member of the Institute of Engineers (IEI) and a member of IEEE. He has several research publications in reputed SCI and Scopus-indexed journals and conferences. He also holds three granted national patents and one published patent.

Amarendra K, Awarded Ph.D(Computer Science & Engineering) from GITAM University, Visakhapatnam, in 2007. With extensive academic experience, he served as Head of the Department and Principal. He is currently working as a Professor and Head of the Department in Computer Science and Engineering at KL University (Koneru Lakshmaiah Education Foundation), Andhra Pradesh, India. His research interests include the Internet of Things (IoT), Machine Learning (ML), Software Engineering(SE) and Deep Learning(DL). He is a lifetime member of the Institute of Engineers (IEI), Life Member of ISTE (Indian Society for Technical Education), Life Member of Indian Red Cross Society, Member of IAENG (International Association of Engineers), Member of CSTA (Computer Science Teachers Association) and a member of IEEE. He has several research publications in

reputed SCI and Scopus-indexed journals and conferences.

How to cite this paper: Krishna Bhimaavarapu, Amarendra K., "Deep Learning based Triple Task Learning Framework for COVID-19 Severity Detection Using CT Images", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.17, No.5, pp. 76-97, 2025. DOI:10.5815/ijigsp.2025.05.06