A Novel Convolutional Neural Network-Based Segmentation Model for Lung CT Scan Images Affected by COVID-19

Varun Srivastava, Nikhil Kalra, Ayushi Tulsyan, and Romy Kumari

Abstract In recent times, the detection of COVID by lung CT scan images has become an active field of research due to the increase in the number of COVID cases worldwide. COVID causes lesion-based damage in the lungs which can be easily analyzed by a CT scan image. The proposed methodology uses a publicly available database of lung computed tomography (CT) scan images collected from 297 subjects having 8739 scans and thereby apply a Covi-Net model for lesion-based segmentation and thereby COVID detection. The Covi-Net model is an extension of U-Net model used for biomedical image classification. The model outperformed related algorithms with a dice value of 0.886.

Keywords COVID-19 detection · Biomedical image classification · Dice coefficient · COVID-19

1 Introduction

In December 2019, a novel respiratory tract disease called COVID-19 was found which was caused due to a virus called SARS-COV-2. Since the virus was able to spread very swiftly by human to human transmission, the situation has been declared a pandemic in January 2020 [1].

Till now, reverse-transcription-polymerase chain reaction (RT-PCR) is one of the most common diagnostic methods for detecting nucleotides from materials obtained by oropharyngeal swab, nasopharyngeal swab, bronchoalveolar lavage, or tracheal aspirate [2, 3]. However, recent findings have indicated that RT-PCR's sensitivity in the detection of COVID-19 has been poor [4, 5]. This could be due to specimen quality or lack of viral material in the sample being tested. COVID-19 patients' chest's scan

V. Srivastava (✉)
Thapar Institute of Engineering and Technology, Patiala, Punjab, India
e-mail: varun.srivastava@thapar.edu

N. Kalra · A. Tulsyan · R. Kumari
Bharati Vidyapeeth's College of Engineering, New Delhi, India

P. K. Singh et al. (eds.), *Futuristic Trends in Networks and Computing Technologies*,
Lecture Notes in Electrical Engineering 936,
https://doi.org/10.1007/978-981-19-5037-7_60

images, on the other hand, frequently display bilateral patchy shadows or ground glass opacity in the lung [6], hence CT has become an important component of the diagnostic process.

However, with increase in confirmed and probable COVID-19 cases, the time taking process of physically contouring lung lesions has become obsolete. Building a quick automated segmentation for COVID-19 infection is critical for illness evaluation to speed up diagnosis and improve access to therapy.

Wang et al. [15] created an algorithm for extracting COVID-19's graphical features and providing a clinical diagnosis before the pathogenic test.

To discover COVID-19, Ayrton [16] used the transfer learning technique using the ResNet50 backbone. Wang et al. [14] proposed a DCNN that was built to recognize COVID-19 patients from tomography images. Multiple deep learning approaches have been explained in [17] for the prognosis of COVID-19. Gozes et al. [18] introduced a system that combines 2D and 3D deep-NN models with clinical knowledge. They modified and extended existing deep network models.

Tang et al. [19] quantify severity by using quantitative analysis through random forest algorithm. For classification, Shi et al. [20] proposed SA random forest technique to combat infection (iSARF). In [21], authors created a method based on deep learning for segmenting and quantifying infection areas from chest images. Deep learning models have clearly shown a major role in diagnosis of diseases in biomedical CT scans. Some of the techniques are given in [7–13].

Further, many deep learning pedagogies have been proposed by various researchers for the prediction and segmentation of COVID-19 in lung biomedical images. In [22], authors trained a ResNet model for the classification of X-Ray images of COVID patients taken from multiple sources. The authors claimed accuracy of around 98%.

In [23], nine architectures, viz. Baseline CNN, DenseNet201, VGG16, VGG19, Inception_ResNet_V2, Inception V3, Xception, ResNet50, and MobileNet_V2 have been used for the classification of X-Rays with Pnemonia and COVID-19. All the architectures have been compared and MobileNet_V2, Inception_Resnet_V2, and ResNet50 were found to be performing better than 96%. Authors in [24] used deep learning to distinguish Coronavirus from simple influenza (a viral fever/pneumonia). A live dataset has been used from three hospitals in Zhejiang province of China with 618 sample images. The architecture achieved an accuracy of around 86.7%.

Authors in [25] introduced a COVIDX-Net model for diagnosing COVID-19 using X-Ray images. The classifier has been built on seven deep learning models, viz. DenseNet121, Xception, Inception V3, Inception-ResNet-V2, ResNet-V2, etc. Out of all these models, VGG19 and DenseNet outperformed with 90 f1-score in average for both classes—healthy and COVID—affected.

In [26], ResNet50, Inception-ResNet-V2, and Inception V3 model have been implemented for a dataset collected from 50 COVID patients and 50 normal ones. ResNet outperformed all other models in terms of accuracy for COVID detection. Authors in [27] have used 300 CT images to first segment the image for COVID-affected area and then cropped the affected area. Four subsets of patches have been created and features like gray-level cooccurence matrix, gray-level size zone matrix,

local directional pattern, last discrete wavelet transform, etc., have been extracted. An accuracy of around 99.68% has been achieved through this.

In [28], authors used deep learning models like VGG16 and VGG19. on images from GitHub repository with images from 75 subjects (50 COVID-19 and 25 healthy) have been used for segmentation followed by classification of images using support vector machine (SVM) model. The model with SVM and ResNet50 performed the best with 95.52% accuracy.

Few more techniques for segmentation and detection of COVID-19 in patients are summarized in [29–34].

In this research, we attempt to develop a new customized deep COVID-19-infected chest CT image detection model. The images are first segmented using a convolutional neural network (CNN). Groundglass opacities (GGOs), areas of consolidation, and a combination of both can be seen in all lung lobes in the chest CT images with COVID-19 infection as shown in Fig. 1. The most of lesions were observed in the periphery, with a modest preference for dorsal lung locations.

The boundaries of a lung CT scan image are hard to distinguish from the chest wall, making their segmentation difficult. Annotation processing involves adjusting various characteristics such as window width and window locations. As shown in Fig. 1, this adjustment reveals the limits of COVID-19 infection zones, which further leads to COVID-19 infection picture segmentation.

The primary contributions of the proposed research work are summarized as follows:

- For the segmentation infected areas from CT images, a unique DL network (Covi-Net) is proposed. Covi-Net is a three-dimensional (3D) convolutional DL system that uses chest images to autonomously segment COVID-19 infected spots and the overall lung. To compute the dice coefficient (DC), a model is suggested to segment the infection mask from the chest image and thereby compute various loss functions followed by boundary surface loss function.

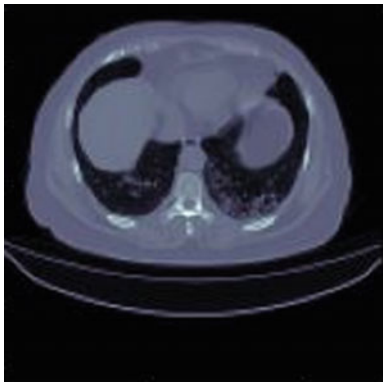

Fig. 1 Chest CT image of a patient suffering with COVID-19

- The problem of boundary loss during segmentation is addressed, which is a major issue in the detection of COVID-19 infection regions, is addressed using a surface loss function.

The paper has been arranged in the following manner. Section 2 describes the methodology and step-by-step algorithm for the segmentation of the COVID-19-affected region. Section 3 summarizes the results and establishes the superiority of the proposed work over related methodologies. Section 4 presents the conclusion and future scope.

2 Methodology

A. Dataset

A publicly available dataset from [35] has been used which contains 349 COVID-19 CT images from 216 patients and 463 non-COVID-19 CTs.

B. Proposed Algorithm

(1) Preprocessing

The input image is first normalized. The pixel intensities that do not have major information are cropped. This is done by using Hounsfield Units (HU) transform. Hounsfield transform computes the CT units in an interpretable format for easy extraction of relevant information from an image. HU transform is a scale to measure the radiodensity of CT scan images. In CT scan images, the HU value is also termed as CT number.

Thereby the pixels having maximum information can be plotted against Housfield Units. Figure 2 shows the various pixel intensity values in CT images. The intensity range from -400 to 600 had the maximum information, and therefore, all other intensity values are trimmed.

(2) Architecture

Covi-Net is a semantic segmentation architecture used for the segmentation of the COVID-19 imges. It is a fine-tuned U-Net model used for classification. In the traditional U-Net model, more convolutional and pooling layers are added with stride as 2 and filter size as 2×2 to obtain the Covi-Net model.

Figure 3 presents the Covi-Net model with layers and output at each layer used in the proposed methodology. It has two 3×3 convolutions separated by a 2×2 max-pooling layer with stride-value as 2 and a corrected linear unit (ReLU). The number of feature channels was raised by four times with each down-sampling step. A 2×2 convolution layer reduces feature channel to half, followed by a merger with feature map from diminishing path. Then two layers with kernel size 3×3, each having ReLU in expansive path are used for further segmentation. Total parameters used have been 35,238,293.

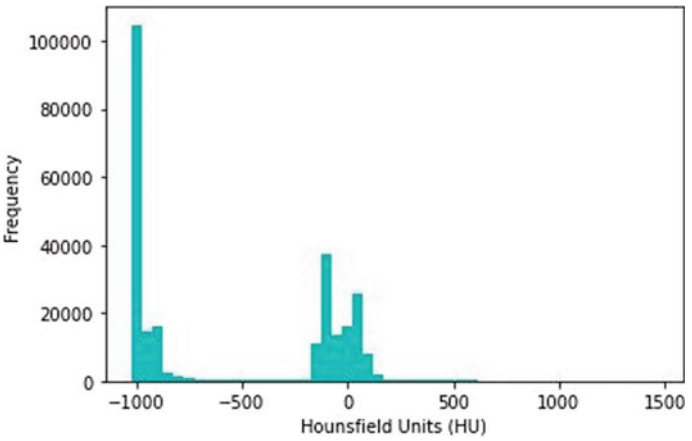

Fig. 2 Hounsfield units (HU) of the input CT scan image of lungs with COVID-19

To avoid border pixel loss that occur after every layer, cropping is essential. The last layer uses 1×1 convolution to convert each 64-component maps to the number of output classes. The network comprises a total of 23 convolutional layers.

(3) **Computation of Loss Functions**

The following loss functions were used:

1. Dice Loss: The Term Dice Loss Comes from the Srensen–Dice Coefficient, Which is Used to Compare Two Samples as Given in Eq. (1).

$$\mathrm{DC} = \frac{2\sum_{i}^{n} p_i * g_i}{\sum_{i}^{n} p_i^2 * \sum_{i}^{n} g_i^2} \tag{1}$$

Here p_i and g_i are pixel pairs with predicted and ground truth values.

(4) **Segmentation**

On the test set, the Covi-Net's performance evaluation has been done for the lung segmentation task. After segmentation, the findings are quite near to being manually annotated. On comparison with similar approaches, the U-Net+ + approach frequently misses the lung's boundary. The VNet approach is unable to provide a smooth lung segmentation boundary, but the proposed model pass the test without any difficulties.

As demonstrated in Fig. 4, the intensity of COVID-19 infection areas is remarkably similar to that of the lung. The areas damaged due to COVID-19 have been highlighted with a purple marker. Also Fully CN, U-Net, and Virtual-Net results in boundary

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 128, 128, 1)	0	
conv2d (Conv2D)	(None, 128, 128, 64)	640	input_1[0][0]
activation (Activation)	(None, 128, 128, 64)	0	conv2d[0][0]
conv2d_1 (Conv2D)	(None, 128, 128, 64)	36928	activation[0][0]
activation_1 (Activation)	(None, 128, 128, 64)	0	conv2d_1[0][0]
max_pooling2d (MaxPooling2D)	(None, 64, 64, 64)	0	activation_1[0][0]
conv2d_2 (Conv2D)	(None, 64, 64, 128)	73856	max_pooling2d[0][0]
batch_normalization (BatchNorma	(None, 64, 64, 128)	512	conv2d_2[0][0]
activation_2 (Activation)	(None, 64, 64, 128)	0	batch_normalization[0][0]
conv2d_3 (Conv2D)	(None, 64, 64, 128)	147584	activation_2[0][0]
batch_normalization_1 (BatchNor	(None, 64, 64, 128)	512	conv2d_3[0][0]
activation_3 (Activation)	(None, 64, 64, 128)	0	batch_normalization_1[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 128)	0	activation_3[0][0]
conv2d_4 (Conv2D)	(None, 32, 32, 256)	295168	max_pooling2d_1[0][0]
batch_normalization_2 (BatchNor	(None, 32, 32, 256)	1024	conv2d_4[0][0]
activation_4 (Activation)	(None, 32, 32, 256)	0	batch_normalization_2[0][0]
conv2d_5 (Conv2D)	(None, 32, 32, 256)	590080	activation_4[0][0]
batch_normalization_3 (BatchNor	(None, 32, 32, 256)	1024	conv2d_5[0][0]
activation_5 (Activation)	(None, 32, 32, 256)	0	batch_normalization_3[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 256)	0	activation_5[0][0]
conv2d_6 (Conv2D)	(None, 16, 16, 512)	1180160	max_pooling2d_2[0][0]
batch_normalization_4 (BatchNor	(None, 16, 16, 512)	2048	conv2d_6[0][0]
activation_6 (Activation)	(None, 16, 16, 512)	0	batch_normalization_4[0][0]
conv2d_7 (Conv2D)	(None, 16, 16, 512)	2359808	activation_6[0][0]
batch_normalization_5 (BatchNor	(None, 16, 16, 512)	2048	conv2d_7[0][0]
activation_7 (Activation)	(None, 16, 16, 512)	0	batch_normalization_5[0][0]
dropout (Dropout)	(None, 16, 16, 512)	0	activation_7[0][0]
max_pooling2d_3 (MaxPooling2D)	(None, 8, 8, 512)	0	dropout[0][0]
conv2d_8 (Conv2D)	(None, 8, 8, 1024)	4719616	max_pooling2d_3[0][0]
batch_normalization_6 (BatchNor	(None, 8, 8, 1024)	4096	conv2d_8[0][0]
activation_8 (Activation)	(None, 8, 8, 1024)	0	batch_normalization_6[0][0]

Fig. 3 Various layers of the Covi-Net model, Output at each layer and the next layer connected to the previous layer

conv2d_9 (Conv2D)	(None, 8, 8, 1024)	9438208	activation_8[0][0]
batch_normalization_7 (BatchNor	(None, 8, 8, 1024)	4096	conv2d_9[0][0]
activation_9 (Activation)	(None, 8, 8, 1024)	0	batch_normalization_7[0][0]
dropout_1 (Dropout)	(None, 8, 8, 1024)	0	activation_9[0][0]
up_sampling2d (UpSampling2D)	(None, 16, 16, 1024)	0	dropout_1[0][0]
conv2d_10 (Conv2D)	(None, 16, 16, 512)	4719104	up_sampling2d[0][0]
batch_normalization_8 (BatchNor	(None, 16, 16, 512)	2048	conv2d_10[0][0]
activation_10 (Activation)	(None, 16, 16, 512)	0	batch_normalization_8[0][0]
conv2d_11 (Conv2D)	(None, 16, 16, 512)	262656	activation_10[0][0]
conv2d_12 (Conv2D)	(None, 16, 16, 512)	262656	dropout[0][0]
batch_normalization_9 (BatchNor	(None, 16, 16, 512)	2048	conv2d_11[0][0]
batch_normalization_10 (BatchNo	(None, 16, 16, 512)	2048	conv2d_12[0][0]
add (Add)	(None, 16, 16, 512)	0	batch_normalization_9[0][0] batch_normalization_10[0][0]
activation_11 (Activation)	(None, 16, 16, 512)	0	add[0][0]
conv2d_13 (Conv2D)	(None, 16, 16, 1)	513	activation_11[0][0]
batch_normalization_11 (BatchNo	(None, 16, 16, 1)	4	conv2d_13[0][0]
activation_12 (Activation)	(None, 16, 16, 1)	0	batch_normalization_11[0][0]
multiply (Multiply)	(None, 16, 16, 512)	0	dropout[0][0] activation_12[0][0]
concatenate (Concatenate)	(None, 16, 16, 1024)	0	activation_10[0][0] multiply[0][0]
conv2d_14 (Conv2D)	(None, 16, 16, 512)	4719104	concatenate[0][0]
activation_13 (Activation)	(None, 16, 16, 512)	0	conv2d_14[0][0]
conv2d_15 (Conv2D)	(None, 16, 16, 512)	2359808	activation_13[0][0]
activation_14 (Activation)	(None, 16, 16, 512)	0	conv2d_15[0][0]
up_sampling2d_1 (UpSampling2D)	(None, 32, 32, 512)	0	activation_14[0][0]
conv2d_16 (Conv2D)	(None, 32, 32, 256)	1179904	up_sampling2d_1[0][0]
batch_normalization_12 (BatchNo	(None, 32, 32, 256)	1024	conv2d_16[0][0]
activation_15 (Activation)	(None, 32, 32, 256)	0	batch_normalization_12[0][0]
conv2d_17 (Conv2D)	(None, 32, 32, 256)	65792	activation_15[0][0]
conv2d_18 (Conv2D)	(None, 32, 32, 256)	65792	activation_5[0][0]
batch_normalization_13 (BatchNo	(None, 32, 32, 256)	1024	conv2d_17[0][0]

Fig. 3 (continued)

batch_normalization_14 (BatchNo	(None, 32, 32, 256)	1024	conv2d_18[0][0]
activation_16 (Activation)	(None, 32, 32, 256)	0	add_1[0][0]
conv2d_19 (Conv2D)	(None, 32, 32, 1)	257	activation_16[0][0]
batch_normalization_15 (BatchNo	(None, 32, 32, 1)	4	conv2d_19[0][0]
activation_17 (Activation)	(None, 32, 32, 1)	0	batch_normalization_15[0][0]
multiply_1 (Multiply)	(None, 32, 32, 256)	0	activation_5[0][0] activation_17[0][0]
concatenate_1 (Concatenate)	(None, 32, 32, 512)	0	activation_15[0][0] multiply_1[0][0]
conv2d_20 (Conv2D)	(None, 32, 32, 256)	1179904	concatenate_1[0][0]
activation_18 (Activation)	(None, 32, 32, 256)	0	conv2d_20[0][0]
conv2d_21 (Conv2D)	(None, 32, 32, 256)	590080	activation_18[0][0]
activation_19 (Activation)	(None, 32, 32, 256)	0	conv2d_21[0][0]
up_sampling2d_2 (UpSampling2D)	(None, 64, 64, 256)	0	activation_19[0][0]
conv2d_22 (Conv2D)	(None, 64, 64, 128)	295040	up_sampling2d_2[0][0]
batch_normalization_16 (BatchNo	(None, 64, 64, 128)	512	conv2d_22[0][0]
activation_20 (Activation)	(None, 64, 64, 128)	0	batch_normalization_16[0][0]
conv2d_23 (Conv2D)	(None, 64, 64, 128)	16512	activation_20[0][0]
conv2d_24 (Conv2D)	(None, 64, 64, 128)	16512	activation_3[0][0]
batch_normalization_17 (BatchNo	(None, 64, 64, 128)	512	conv2d_23[0][0]
batch_normalization_18 (BatchNo	(None, 64, 64, 128)	512	conv2d_24[0][0]
add_2 (Add)	(None, 64, 64, 128)	0	batch_normalization_17[0][0] batch_normalization_18[0][0]
activation_21 (Activation)	(None, 64, 64, 128)	0	add_2[0][0]
conv2d_25 (Conv2D)	(None, 64, 64, 1)	129	activation_21[0][0]
batch_normalization_19 (BatchNo	(None, 64, 64, 1)	4	conv2d_25[0][0]
activation_22 (Activation)	(None, 64, 64, 1)	0	batch_normalization_19[0][0]
multiply_2 (Multiply)	(None, 64, 64, 128)	0	activation_3[0][0] activation_22[0][0]
concatenate_2 (Concatenate)	(None, 64, 64, 256)	0	activation_20[0][0] multiply_2[0][0]
conv2d_26 (Conv2D)	(None, 64, 64, 128)	295040	concatenate_2[0][0]
activation_23 (Activation)	(None, 64, 64, 128)	0	conv2d_26[0][0]
conv2d_27 (Conv2D)	(None, 64, 64, 128)	147584	activation_23[0][0]
activation_24 (Activation)	(None, 64, 64, 128)	0	conv2d_27[0][0]

Fig. 3 (continued)

up_sampling2d_3 (UpSampling2D)	(None, 128, 128, 128 0	activation_24[0][0]
conv2d_28 (Conv2D)	(None, 128, 128, 64) 73792	up_sampling2d_3[0][0]
batch_normalization_20 (BatchNo	(None, 128, 128, 64) 256	conv2d_28[0][0]
activation_25 (Activation)	(None, 128, 128, 64) 0	batch_normalization_20[0][0]
conv2d_29 (Conv2D)	(None, 128, 128, 64) 4160	activation_25[0][0]
conv2d_30 (Conv2D)	(None, 128, 128, 64) 4160	activation_1[0][0]
batch_normalization_21 (BatchNo	(None, 128, 128, 64) 256	conv2d_29[0][0]
batch_normalization_22 (BatchNo	(None, 128, 128, 64) 256	conv2d_30[0][0]
add_3 (Add)	(None, 128, 128, 64) 0	batch_normalization_21[0][0] batch_normalization_22[0][0]
activation_26 (Activation)	(None, 128, 128, 64) 0	add_3[0][0]
conv2d_31 (Conv2D)	(None, 128, 128, 1) 65	activation_26[0][0]
batch_normalization_23 (BatchNo	(None, 128, 128, 1) 4	conv2d_31[0][0]
activation_27 (Activation)	(None, 128, 128, 1) 0	batch_normalization_23[0][0]
multiply_3 (Multiply)	(None, 128, 128, 64) 0	activation_1[0][0] activation_27[0][0]
concatenate_3 (Concatenate)	(None, 128, 128, 128 0	activation_25[0][0] multiply_3[0][0]
conv2d_32 (Conv2D)	(None, 128, 128, 64) 73792	concatenate_3[0][0]
activation_28 (Activation)	(None, 128, 128, 64) 0	conv2d_32[0][0]
conv2d_33 (Conv2D)	(None, 128, 128, 64) 36928	activation_28[0][0]
activation_29 (Activation)	(None, 128, 128, 64) 0	conv2d_33[0][0]
conv2d_34 (Conv2D)	(None, 128, 128, 1) 65	activation_29[0][0]

Fig. 3 (continued)

loss. However, the proposed methods produced excellent results with better DSC score, similar to manual annotation. The various steps of the proposed technique are summarized in Fig. 5.

3 Results and Discussions

The images obtained after preprocessing of the original image are given in Fig. 6. The preprocessing involves computation of a normalized image by using Hounsfield transform and thereby selecting the pixels in the range having maximum information.

The dice coefficient (DC) has been used for evaluating and comparing the segmentation results of the models, which is defined as given in Eq. (2).

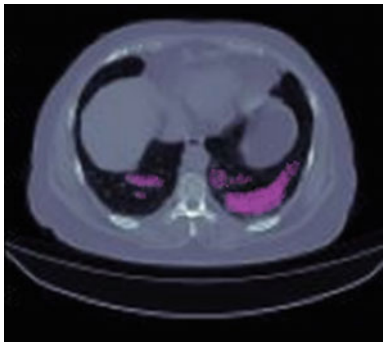

Fig. 4 Segmented image obtained by Covi-Net model

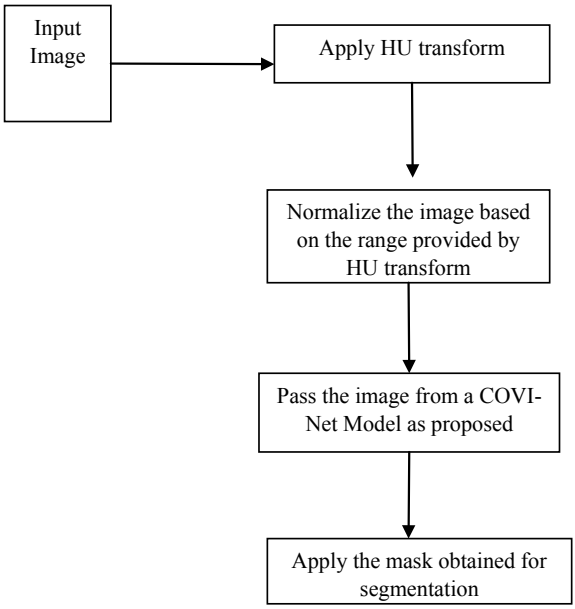

Fig. 5 Steps involved in the segmentation of lung CT scan images using Covi-Net model

$$\mathrm{DSC} = \frac{2|X \cap Y|}{|X| + |Y|} \tag{2}$$

where X is the model's semantic segmentation output and Y is an expert-specified mask of infected regions. The DSC scale runs from 0 to 1, with 1 indicating the highest degree of similarity between the model output (prediction) and the truth ground. Table 1 gives the DSC and cross-entropy loss acquired for each model for the testing set. According to these findings, Covi-Net outperformed the other models, indicating that attention blocks and recurring routes may be important in detecting lesions.

The loss decreases with each epoch, is shown in Fig. 7. The appropriate training of the model is depicted by a constant decrease in loss value with each epoch and a flat

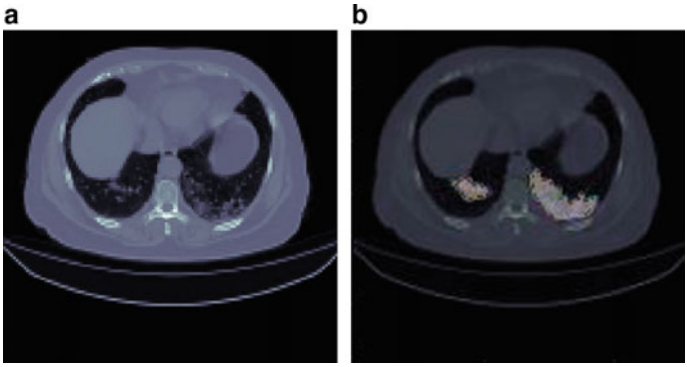

Fig. 6 **a** Original image and **b** normalized image

Table 1 Comparison of Covi-NET model with existing similar achitectures

Model	DSC
U-Net [36]	0.77
R2 U-Net [37]	0.76
Attention U-Net [38]	0.77
Attention R2 U-Net	0.79
Covi-Net	0.88

stabilizing curve at the conclusion. Figure 7 shows that the loss function's minimal value has been reached.

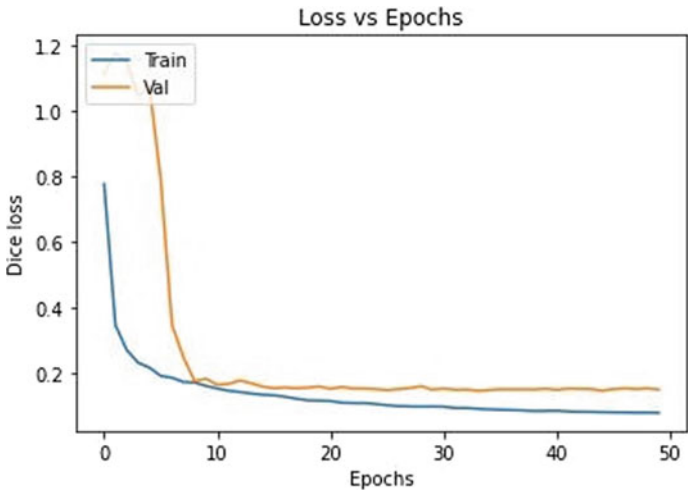

Fig. 7 Dice loss with each epoch during training of segmentation model

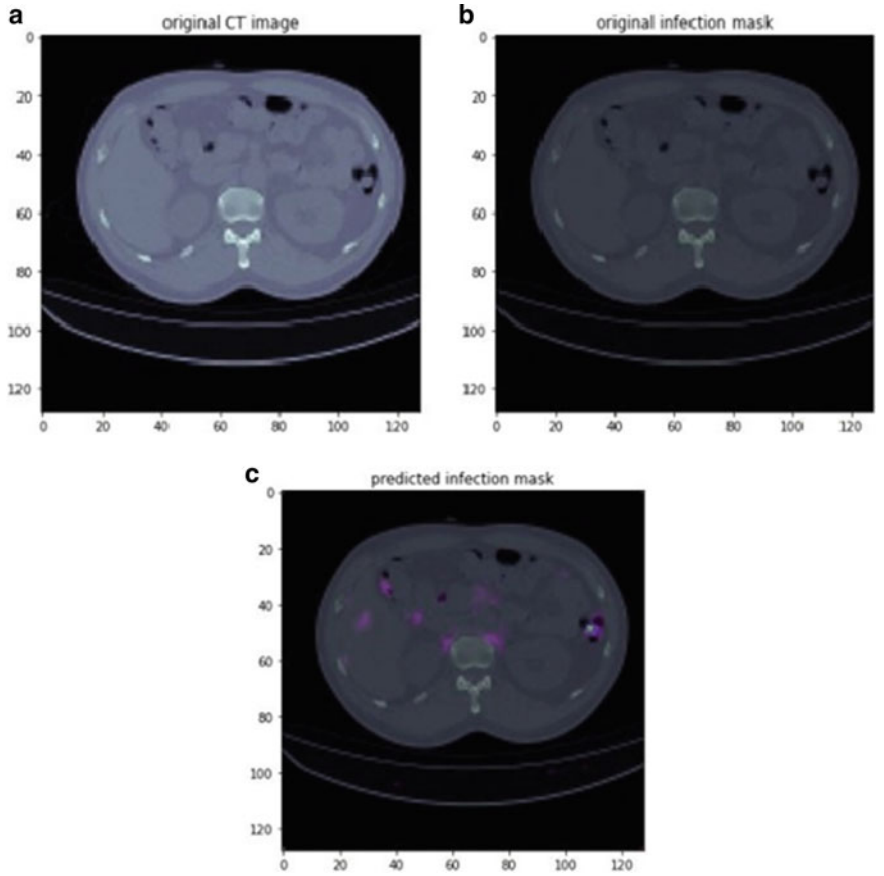

Fig. 8 **a** Orignal CT image, **b** original infection mask, and **c** predicted infection mask by Covi-Net

The prediction of lesion volumes to CT volumes has been tested to further evaluate the model's accuracy and compare the segmentation findings to the ground truth. The results in Fig. 8 reveal that the lesion volumes recognized by the model are strongly correlated with those discovered by the expert, indicating that the proposed model has a high power of prediction. The lesions have been highlighted with a purple marker and indicates the damage done due to COVID-19.

4 Conclusion and Future Scope

As observed from Table 1, the proposed Covi-Net model outperforms other state-of-the-art techniques. It has improved the DSC value for segmented images by 12.5%,

13.6%, 12.5%, and 10.2% over U-Net, R2-U-Net, Attention U-Net, and Attention R2-U-Net, respectively.

The proposed CNN framework has been obtained by fine-tuning U-Net's existing architecture. Similarly, other CNN networks like Alex-Net, VGG Models, Google-Net, etc., can be fine-tuned to achieve better segmentation models.

References

1. Bedford J, Enria D, Giesecke J, Heymann DL, Ihekweazu C, Kobinger G, Wieler LH (2020) Living with the COVID-19 pandemic: act now with the tools we have. The Lancet 396(10259):1314–1316
2. Silva ALOD, Moreira JC, Martins SR (2020) COVID-19 and smoking: a high-risk association. Cad Saude Publica 36:e00072020
3. Bai HX, Hsieh B, Xiong Z, Halsey K, Choi JW, Tran TML, Liao WH (2020) Performance of radiologists in differentiating COVID-19 from non-COVID-19 viral pneumonia at chest CT. Radiology 296(2):E46–E54
4. Ai T, Yang Z, Hou H, Zhan C, Chen C, Lv W, Xia L (2020) Correlation of chest CT and RT-PCR testing for coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases. Radiology 296(2):E32–E40
5. Fang Y, Zhang H, Xie J, et al (2020) Sensitivity of chest CT for COVID-19: comparison to RT-PCR. Radiology, 200432
6. Wang D, Hu B, Hu C, Zhu F, Liu X, Zhang J, Peng Z (2020) Clinical characteristics of 138 hospitalized patients with 2019 novel coronavirus–infected pneumonia in Wuhan, China. Jama 323(11):1061–1069
7. Yan Q, Zhang L, Liu Y, Zhu Y, Sun J, Shi Q, Zhang Y (2020) Deep HDR imaging via a non-local network. IEEE Trans Image Process 29:4308–4322
8. Yan Q, Gong D, Zhang Y (2018) Two-stream convolutional networks for blind image quality assessment. IEEE Trans Image Process 28(5):2200–2211
9. Yan Q, Gong D, Zhang P, Shi Q, Sun J, Reid I, Zhang Y (2019) Multi-scale dense networks for deep high dynamic range imaging. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp 41–50, January. IEEE
10. Yan Q, Gong D, Shi Q, Hengel AVD, Shen C, Reid I, Zhang Y (2019) Attention-guided network for ghost-free high dynamic range imaging. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 1751–1760
11. Gong D, Yang J, Liu L, Zhang Y, Reid I, Shen C, et al (2017) From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2319–2328
12. He T, Shen C, Tian Z, Gong D, Sun C, Yan Y (2019) Knowledge adaptation for efficient semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 578–587
13. Gong D, Liu L, Le V, Saha B, Mansour MR, Venkatesh S, Hengel AVD (2019) Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 1705–1714
14. Wang L, Lin ZQ, Wong A (2020) Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. Sci Rep 10(1):1–12
15. Wang S, Kang B, Ma J, Zeng X, Xiao M, Guo J, et al (2021) A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19). European Radiol, 1–9
16. Joaquin AS, To detect pneumonia caused by NCOV- 19 from x-ray images. https://towardsdatascience.com/using-deep-learning-to-detect-ncov-19-from-x-ray-images-1a89701d1acd. Accessed November 2, 2021

17. Chowdhury MEH, Rahman T, Khandakar A, Mazhar R, Kadir MA, Mahbub ZB, Islam KR et al (2020) Can AI help in screening viral and COVID-19 pneumonia? IEEE Access 8:132665–132676
18. Gozes O, Frid-Adar M, Greenspan H, Browning PD, Zhang H, Ji W, Siegel E (2020) Rapid Al development cycle for the coronavirus (covid-19) pandemic: Initial results for automated detection & patient monitoring using deep learning CT image analysis. arXiv preprint arXiv:2003.05037
19. Tang Z, Zhao W, Xie X, Zhong Z, Shi F, Liu J, Shen D (2020) Severity assessment of coronavirus disease 2019 (COVID-19) using quantitative features from chest CT images. arXiv preprint arXiv:2003.11988
20. Shi F, Xia L, Shan F, Song B, Wu D, Wei Y, Shen D (2021) Large-scale screening to distinguish between COVID-19 and community-acquired pneumonia using infection size-aware classification. Phys Med Biol 66(6):065031
21. Shan F, Gao Y, Wang J, Shi W, Shi N, Han M, et al (2020) Lung infection quantification of COVID-19 in CT images with deep learning. arXiv preprint arXiv:2003.04655
22. Apostolopoulos ID, Mpesiana TA (2020) Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. Phys Eng Sci Med 43(2):635–640
23. El Asnaoui K, Chawki Y, Idri A (2021) Automated methods for detection and classification pneumonia based on x-ray images using deep learning. In: Artificial intelligence and blockchain for future cybersecurity applications. Springer, Cham, pp 257–284
24. Xu X, Jiang X, Ma C, Du P, Li X, Lv S, Li L (2020) A deep learning system to screen novel coronavirus disease 2019 pneumonia. Engineering 6(10):1122–1129
25. Hemdan EED, Shouman MA, Karar ME (2020) Covid x-net: a framework of deep learning classifiers to diagnose covid-19 in x-ray images. arXiv preprint arXiv:2003.11055
26. Department of Biomedical Engineering, Zonguldak Bulent Ecevit University, 67100, Zonguldak, Turkey
27. Barstugan M, Ozkaya U, Ozturk S (2020) Coronavirus (covid-19) classification using CT images by machine learning methods. arXiv preprint arXiv:2003.09424
28. Sethy PK, Behera SK (2020) Detection of coronavirus disease (covid-19) based on deep features
29. Ismael AM, Şengür A (2021) Deep learning approaches for COVID-19 detection based on chest X-ray images. Expert Syst Appl 164:114054
30. Khan AI, Shah JL, Bhat MM (2020) CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images. Comput Methods Programs Biomed 196:105581
31. Brunese L, Mercaldo F, Reginelli A, Santone A (2020) Explainable deep learning for pulmonary disease and coronavirus COVID-19 detection from X-rays. Comput Methods Programs Biomed 196:105608
32. Maghdid HS, Asaad AT, Ghafoor KZ, Sadiq AS, Mirjalili S, Khan MK (2021) Diagnosing COVID-19 pneumonia from X-ray and CT images using deep learning and transfer learning algorithms. In: Multimodal Image Exploitation and Learning (vol 11734). International Society for Optics and Photonics, p. 117340E
33. Gupta A, Gupta S, Katarya R (2021) InstaCovNet-19: A deep learning classification model for the detection of COVID-19 patients using Chest X-ray. Appl Soft Comput 99:106859
34. Bharati S, Podder P, Mondal M, Prasath VB (2021) Medical imaging with deep learning for COVID-19 diagnosis: a comprehensive review. arXiv preprint arXiv:2107.09602
35. Yang X, He X, Zhao J, Zhang Y, Zhang S, Xie P (2020) COVID-CT-dataset: a CT scan dataset about COVID-19. arXiv preprint arXiv:2003.13865
36. Chen X, Yao L, Zhang Y (2020) Residual attention u-net for automated multi-class segmentation of covid-19 chest CT images. arXiv preprint arXiv:2004.05645
37. Zhou T, Canu S, Ruan S (2021) Automatic COVID-19 CT segmentation using U-Net integrated spatial and channel attention mechanism. Int J Imaging Syst Technol 31(1):16–27
38. Oktay O, Schlemper J, Folgoc LL, Lee M, Heinrich M, Misawa K, Rueckert D (2018) Attention u-net: learning where to look for the pancreas. arXiv preprint arXiv:1804.03999