Received: 30 December 2021 | Revised: 27 June 2022 | Accepted: 18 July 2022

DOI: 10.1002/mp.15882

RESEARCH ARTICLE

MEDICAL PHYSICS

COVID-19 CT image segmentation based on improved Res2Net

Shangwang Liu[1,2] | **Xiufang Tang**[1] | **Tongbo Cai**[1] | **Yangyang Zhang**[1] | **Changgeng Wang**[1]

[1]School of Computer and Information Engineering, Henan Normal University, Xinxiang, Henan, China

[2]Engineering Lab of Intelligence Business & Internet of Things, Xinxiang, Henan, China

Correspondence
Shangwang Liu, School of Computer and Information Engineering, Henan Normal University, No. 46 in Jianshe East Road, Xinxiang 453000, Henan Province, China.
Email: shwl2012@hotmail.com

Funding information
Henan Province, Grant/Award Number: 21A520022

Abstract

Purpose: Corona virus disease 2019 (COVID-19) is threatening the health of the global people and bringing great losses to our economy and society. However, computed tomography (CT) image segmentation can make clinicians quickly identify the COVID-19-infected regions. Accurate segmentation infection area of COVID-19 can contribute screen confirmed cases.

Methods: We designed a segmentation network for COVID-19-infected regions in CT images. To begin with, multilayered features were extracted by the backbone network of Res2Net. Subsequently, edge features of the infected regions in the low-level feature f_2 were extracted by the edge attention module. Second, we carefully designed the structure of the attention position module (APM) to extract high-level feature f_5 and detect infected regions. Finally, we proposed a context exploration module consisting of two parallel explore blocks, which can remove some false positives and false negatives to reach more accurate segmentation results.

Results: Experimental results show that, on the public COVID-19 dataset, the Dice, sensitivity, specificity, S_α, E_ϕ^{mean}, and mean absolute error (*MAE*) of our method are 0.755, 0.751, 0.959, 0.795, 0.919, and 0.060, respectively. Compared with the latest COVID-19 segmentation model Inf-Net, the Dice similarity coefficient of our model has increased by 7.3%; the sensitivity (Sen) has increased by 5.9%. On contrary, the *MAE* has dropped by 2.2%.

Conclusions: Our method performs well on COVID-19 CT image segmentation. We also find that our method is so portable that can be suitable for various current popular networks. In a word, our method can help screen people infected with COVID-19 effectively and save the labor power of clinicians and radiologists.

KEYWORDS
context exploration, COVID-19 CT image segmentation, edge attention, Res2Net

1 | INTRODUCTION

Corona virus disease 2019 (COVID-19) is acute respiratory infection caused by novel coronavirus. The pandemic COVID-19 is still to have a devastating impact on the people health and well-being of the global population. A key step in fighting against COVID-19 is effective screening of infected patients, which allows the isolation of infected individuals for immediate treatment to mitigate the spread of the virus. Nowadays, the

primary screening method for detecting COVID-19 cases is the reverse transcription polymerase chain reaction (RT-PCR)[1] testing but is a time-consuming, laborious, and complicated manual process. Furthermore, the sensitivity of RT-PCR testing is highly variable[2,3] and the positive rate of RT-PCR decreases over time.[4,5]

Another method that has been used to screen for COVID-19 is radiography examination (e.g., chest X-ray [CXR] or computed tomography [CT] imaging). Guan et al. found that COVID-19 infectors have imaging radiographic abnormalities in CXR and CT images (e.g., ground-glass opacity [GGO], bilateral abnormalities, and interstitial abnormalities).[6] In contrast to CXR, CT screening is very popular due to its three-dimensional view of the lung. On CT images, areas of COVID-19 infection can be distinguished by pulmonary GGO in the early stages of COVID-19, and by solid lung lesions in the later stages.[7] Many research argued that the sensitivity of CT is higher than and RT-PCR for COVID-19 detection.[8–10] Therefore, CT was an effective supplementary method for RT-PCR to detect COVID-19.[11] However, in the clinicians' practice community, CT method routinely relies on the radiologist depiction of the infection areas; this is a highly subjective task, influenced by clinician bias and experiences. Fortunately, automatic segmentation technology can reduce the labor intensity of radiologists, improve the accuracy of COVID-19 diagnosis, and save precious time for the patients. Therefore, it is necessary to assist radiologists in labeling lung infections by using automatic segmentation techniques.

Nowadays, some of CT image segmentation methods have emerged, and convolutional neural network (CNN) method has been a hot topic. Keshani et al. utilized the support vector machine classifier to detect the lung nodule from CT slices.[11] Wang et al.[12] proposed a data-driven model CF-CNN, and it can extract different sensitive features from 3D and 2D images for CT image segmentation. Jiang et al.[13] put forward two multi-resolution residual networks, which can combine image resolution and image characteristics to segment tumors. Tian et al.[14] introduced the background and development of instance segmentation technology and commonly used datasets in this field. Undoubtedly, these CT image segmentation methods mentioned earlier improve the segmentation performance of specific images to a certain extent.

Recently, some methods[15–19] for COVID-19 have been proposed. Wang et al.[19] proposed a modified inception neural network for classifying COVID-19 patients and normal ones. Chen et al.[20] collected 46 096 CT image slices from COVID-19 patients and control patients with other diseases; these CT images were used to train U-Net++[21] to identify COVID-19 patients. Fan et al.22 put forward the COVID-19 pneumonia infection area segmentation network, Inf-Net: leveraging parallel partial decoder to aggregate high-level features, and exploiting reverse attention to enhance expressiveness. Saood et al.[23] utilized two known deep learning networks (SegNet[24] and U-Net[25]) for image tissue classification. Though these previous methods perform well in COVID-19 CT image segmentation, there are still many incorrectly predicted regions (false positive and false negative regions) and many unclear edges. This is because there are several challenging issues in COVID-19 CT images segmentation as follows:

1. The infected regions in COVID-19 CT images are usually very small; and extremely similar to its background area (see Figure 1).
2. COVID-19 training datasets, especially CT images with ground truth (GT), are often too small to train a high-quality deep learning network.

To address issues mentioned earlier, we take the Res2Net as the backbone and elaborate a network Ref-Net to implement automatic COVID-19 CT image segmentation. Literatures[26,27] pointed out that the low-level convolutional layer contains those edge spatial information features, whereas the high-level convolutional layer retains those global semantic information features. Therefore, in the lower layer of the backbone network, we employ edge attention module (EAM) to extract edge features and generate a prediction map with clearer boundaries; in the higher layer, we well-designed attention position module (APM) to acquire the global prediction map, which can detect the whole object more accurately. It is difficult to train a high-quality network because there are few labeled data during training. This leads to false negative and false positive areas in the prediction results. We proposed a multi-scale context exploration module (CEM) based on the context feature learning mechanism to gradually remove false positives and false negatives. Therefore, CEM can solve the problem of insufficient accuracy of network model caused by less training data. All this can ensure that our COVID-19 CT image segmentation results have not only clearer boundaries but also less false predictions.

2 | MATERIALS AND METHODS

The network framework of our COVID-19 CT image segmentation network (Ref-Net) is shown in Figure 2. In Figure 2, we take Res2Net as the backbone network to extract multilayered features f_1,f_2,f_3,f_4,f_5. Because low-level features retain the edge information of the target object, we extended f_2 feature by using an EAM to generate the edge feature map M_e. High level is rich in the global positioning semantic information of the target object, so we input f_5 into the attention positioning

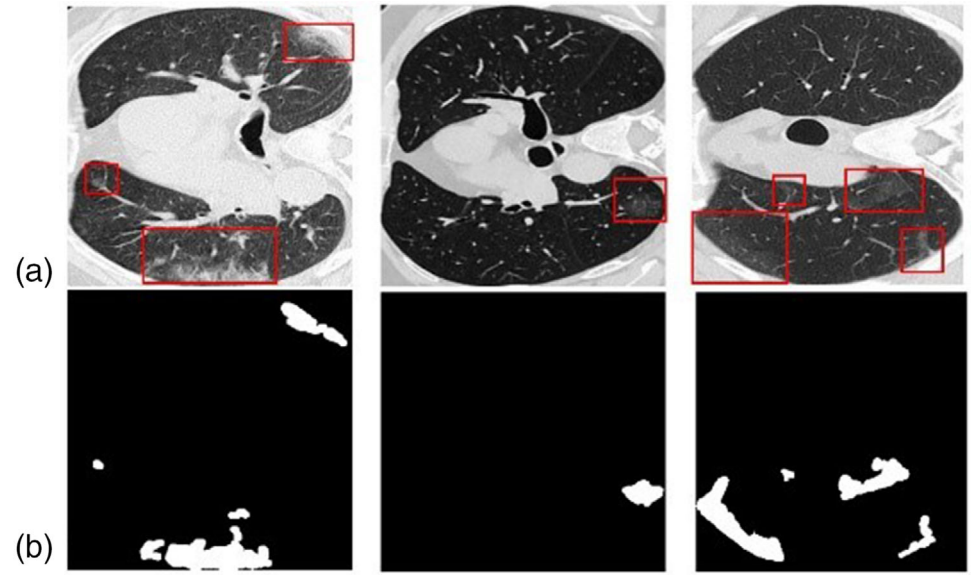

FIGURE 1 Examples of corona virus disease 2019 (COVID-19)-infected regions in computed tomography (CT) (a) and infected regions ground truth (GT) map (b)

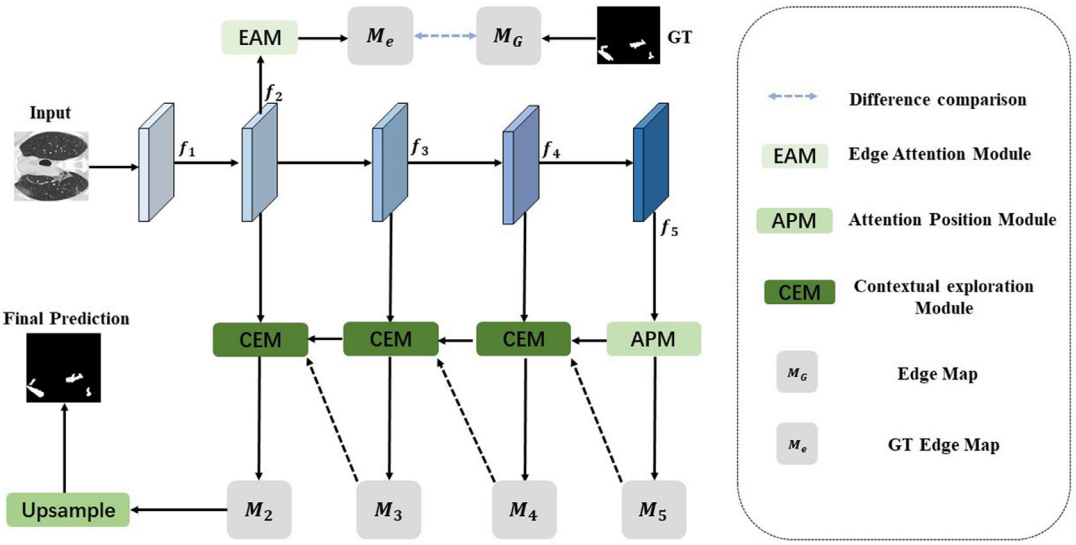

FIGURE 2 Network structure diagram of Ref-Net

module (APM) that is composed of channel attention and spatial attention to get the preliminary prediction map M_5. The three CAMs gradually combine the current features and upper-layer features. In the end, CAM can eliminate false positives and false negatives in the prediction results, and the refined final prediction map is thus obtained.

2.1 | Backbone (Res2Net)

Bottleneck block (see Figure 3a) is the basic building block of many modern backbone CNNs architectures, for example, ResNet,[28] ResNeXt.[29] Res2Net[30] block (see Figure 3b) has expanded the range of the receptive field in each layer and can represent multi-scale features at a finer granularity. Especially, Res2Net replaces the 3 × 3 filters of Figure 3a with smaller filters, while connecting different filter groups in a hierarchical residual-like style.

As shown in Figure 3b, the feature maps are uniformly divided into s subsets of feature maps after 1 × 1 convolution, denoted by x_i, where $i \in \{1, 2, \ldots, s\}$; each x_i (except for x_1) has a corresponding 3 × 3 convolution, denoted by k_i; y_i represents the output of k_i, and it can be expressed by the following

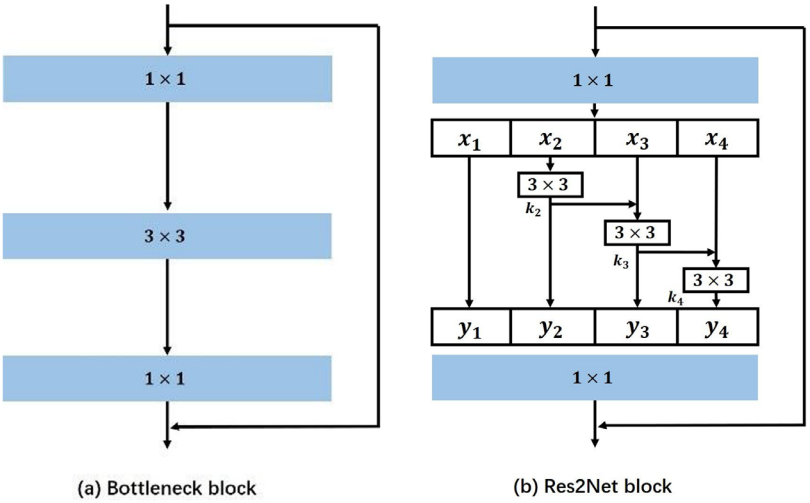

FIGURE 3 The difference between bottleneck block and Res2Net block

equation:

$$y_i = \begin{cases} x_i & i = 1 \\ k_i(x_i) & i = 2 \\ k_i(x_i + y_{i-1}) & 2 < i \le s \end{cases}. \quad (1)$$

Finally, in order to better fuse the information of different scales y_i, all of them are connected to the Res2Net block via a 1 × 1 convolution.

2.2 | Edge attention module (EAM)

From several existing literatures,[31–34] we can see that the edge feature of the infected regions is not taken part in the COVID-19 image segmentation, which is actually the main cause to the blur boundaries. Hence, we introduced the EAM to focus on the edge features of the target. High-level features retain semantic information that can provide abstract descriptions, but they have less verbosely information. Low-level features pay more emphasis on spatial information that can construct object boundaries. Therefore, adding the EAM to the lower layer can better pay attention to the edge features.

We add the EAM after a more appropriate lower layer f_2 to lay emphasis on edge features. Specifically, the EAM is composed of a convolutional layer with one convolution kernel, via which the edge feature map (M_e) can be obtained. At the same time, the true-value edge feature map (M_G) can be generated by using the derivation of the GT. Therefore, the final clearer boundary can be extracted by comparing M_e with M_G to correct M_e iteratively. In detailed implementation, we measured the dissimilarity between the edge feature map (M_e) and the true-value edge feature map (M_G), which is constrained by a standard binary cross-entropy loss function L_e. The loss function could be computed from the following equation:

$$L_e = -\sum_{x=1}^{w}\sum_{y=1}^{h}\left[M_G \log(M_e) + (1 - M_G)\log(1 - M_e)\right], \quad (2)$$

where w and h represent the width and height of the feature map, respectively.

2.3 | Attention position module (APM)

As is well known, attention mechanism can selectively highlight the important areas of an image. Channel attention lays emphasis on which channel features are more meaningful[34,35]; spatial attention pay attention to which area features are more meaningful.[36,37] Combining channel attention and spatial attention can make the network focus on more meaningful features and locate targets more accurately. The structure diagram of APM is shown in Figure 4.

From Figure 4, we can see that the structure diagram of APM is composed of channel attention and spatial attention, and its purpose is to enhance high-level global semantic information and then obtain preliminary segmentation results. The high-layer output f_5 is exactly the input feature F of the APM. The channel number, height, and width of feature F are denoted as C, H and W, respectively. First, the input feature F is reshaped to obtain Q, K, and V (K, Q, and V represent for query, key, and query, respectively, and K, Q, V are abstract concepts used to calculate attention[37]), where $\{Q, K, V\} \in R^{C \times N}$, and N refers to the number of pixels. Matrix multiplication (MM) is used between Q transposed and K, and the result passes through the softmax layer to generate the channel attention map $X \in R^{C \times C}$. The influence of the channel j on the channel i in Figure X is specifically

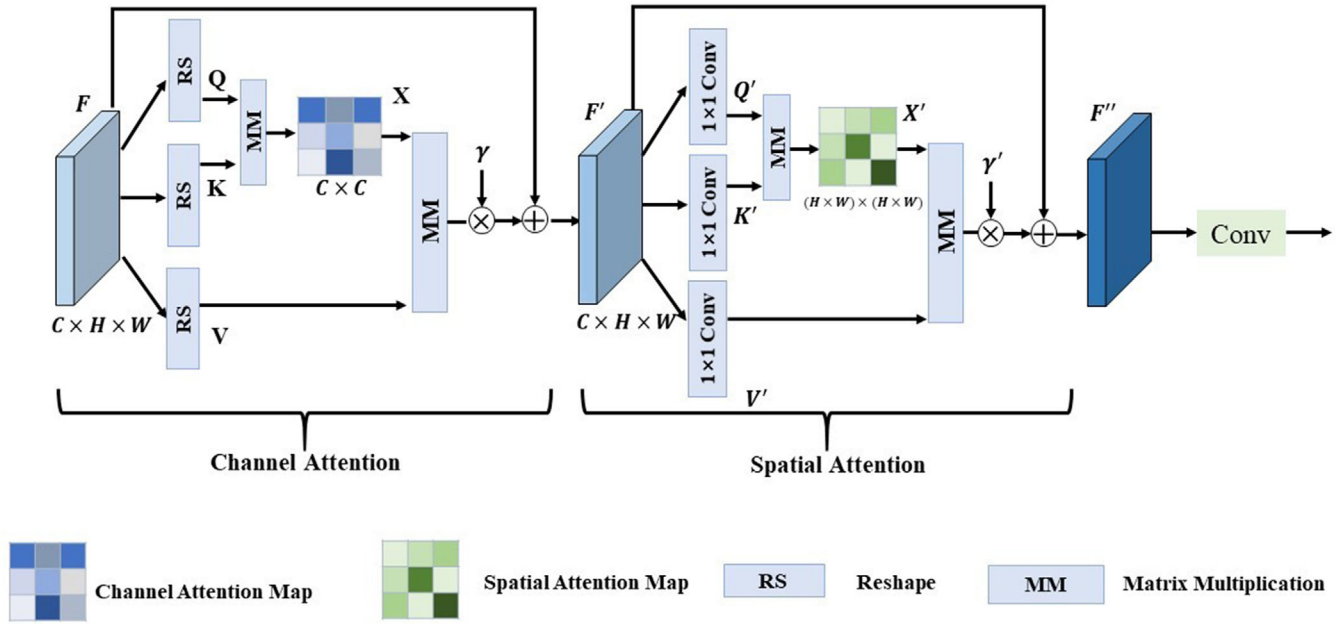

FIGURE 4 Attention position module structure diagram

expressed as the following equation:

$$x_{ij} = \exp\left(\frac{\exp\left(Q_{i:} \cdot K_{j:}\right)}{\sum_{j=1}^{C} \exp\left(Q_{i:} \cdot K_{j:}\right)}\right), \tag{3}$$

where $Q_{i:}$ represents the row i of matrix Q, and $K_{j:}$ represents the row j of matrix K.

Thus, we performed MM between V and X after transposing and resized the resulting feature shape into $R^{C\times H\times W}$. In addition, we also introduced a proportional parameter γ, which is learnable, with an initial value of 1 and constantly learning to update the weight. Finally, the final channel attention output feature F' is obtained by jumping connection. Any line of the output feature F' can be expressed by the following equation:

$$F'_{i:} = \gamma \sum_{j=1}^{C} \left(x_{ij} V_{j:}\right) + F_{i:}, \tag{4}$$

where $F'_{i:}$ represents the row i of the channel attention output feature F'; γ is the proportional parameter; $V_{j:}$ represents the row j of the value V matrix; and $F_{i:}$ indicates the channel attention enter the row i of the feature.

The specific process of spatial attention is similar to channel attention. F' is the input of spatial attention, and it undergoes three 1×1 convolutions and changes the shape to obtain a new query (Q'), key (K'), and value (V'). It is worth noting that $\{Q',\ K'\} \in R^{C_1\times N}$, $C_1 = C/8$, and $V \in R^{C\times N}$. MM exists between the transposition of Q' and K', then it is normalized to get the spatial attention map $X' \in R^{N\times N}$. Different from channel attention, the influence of the position j on the position i in spatial attention is calculated from the following equation:

$$x'_{ij} = \exp\left(\frac{\exp\left(Q'_{:i} \cdot K'_{:j}\right)}{\sum_{j=1}^{N} \exp\left(Q'_{:i} \cdot K'_{:j}\right)}\right), \tag{5}$$

where $Q'_{:i}$ represents the column i of query Q', and $K'_{:j}$ represents the column j of key K'.

The following process is similar to channel attention. The transposition of X' and V' is subjected to MM to change the shape of the result. Introduce the proportional parameter γ' and then jump and connect to obtain the final output F''.

2.4 | Contextual exploration module (CEM)

Contextual feature learning plays an important role in many computers vision tasks, and many works have exploited contextual information that can enhance feature representation.[38–44] That is to say, in the task of image segmentation, context information has strong spatial constraints. Namely, in the process of image segmentation, some ambiguity regions in the prediction results can be removed by context feature information. As is mentioned before, the infected regions of COVID-19 are small and highly similar to the background (see Figure 1a), which always results in false positive

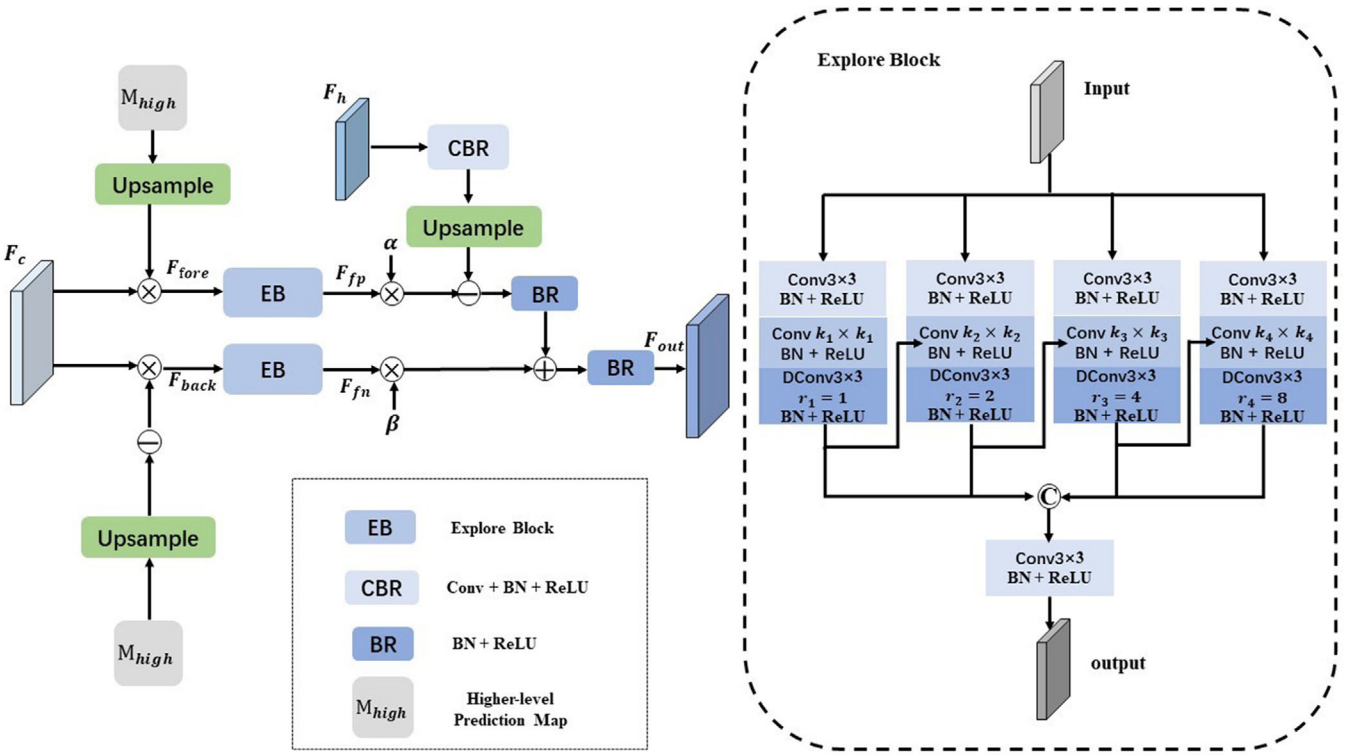

FIGURE 5 Contextual exploration module structure diagram

predictions and false negative predictions in the image segmentation. To solve this problem, we designed a CEM composed of two parallel explore blocks (EBs), so it can find and eliminate both false positive and false negative regions and reach a more accurate segmentation result.

The structure diagram of the CEM is illustrated in Figure 5. As shown in Figure 5 (left), First, we upsampling the high-level prediction maps M_{high} and normalize it with a sigmoid layer. Subsequently, we use this normalized map to multiply the current level feature F_c. Thus, the foreground feature can be further expanded to extract the attention foreground feature F_{fore}. Similarly, we reverse the normalized map to multiply the current level feature F_c to extract the attention background feature F_{back}. F_{fore} and F_{back} are sent into two parallel EB to detect the false positive regions F_{fp} and false negative regions F_{fn} from the prediction results. After detecting the false positive regions F_{fp} and the false negative regions F_{fn}, we can perform as follows:

$$F_1 = U(CBR(F_h)), \tag{6}$$

$$F_2 = BR(F_1 - \alpha F_{fp}), \tag{7}$$

$$F_{out} = BR(F_2 + \beta F_{fn}), \tag{8}$$

where F_h is the upper-level input feature; F_{out} is the output feature after false positives and false negatives are eliminated; C, B, and R represent convolution, normalization, and ReLU, respectively; U represents upsample; both α and β are learnable parameters. The element-wise subtraction of F_1 and αF_{fp} can eliminate false positives, and the element-wise addition of F_2 and βF_{fn} can eliminate false negatives.[43,44]

As shown in Figure 5 (right), the EB consists of four branches, and the structure of the four branches is similar: 3×3 convolution is used for channel reduction; $K_i \times K_i$ convolution is adopted for local feature extraction ($K_1 = 1, K_2 = 3, K_3 = 5, K_4 = 7$). The size of the convolution kernel of the dilated convolution is 3×3, and the expansion rate is r_i, which is used for context awareness. It should be noticed that batch normalization (BN) and ReLU must be performed after each convolution. Each branch must be entered into the next branch for further processing. Finally, the output results of the four branches are superimposed in the channel dimension. The structure of the EB can enrich the context exploration capabilities to discover false positives and false negatives in the prediction results.

2.5 | Loss function

The weighted *IOU* loss function L_{IoU}^{w} is different from other *IOU* loss functions. In order to highlight the

importance of difficult samples, the weight w of difficult sample points is increased. The weighted binary cross-entropy L^w_{BCE} focuses on the difficult sample points to assign weights to them in order to highlight their salience, instead of assigning weights to all pixels in the difficult samples. The definitions of these losses are the same as those ones in Refs. [45, 46], and their effectiveness has been verified. We introduce a loss function to combine L^w_{IoU} and L^w_{BCE} as follows:

$$L_s = L^w_{IoU} + \mu L^w_{BCE}, \tag{9}$$

where μ represent the weight, and we set it to 1; L^w_{IoU} and L^w_{BCE} provide global supervision and local supervision, respectively, and the segmentation results are more accurate.

We also supervise the output map M_5 by using APM and the outputs M_4,M_3,and M_2 of the three CEMs. The loss function could be computed from the following equation:

$$L_i = L_s\left(GT, M_i^{up}\right), \tag{10}$$

where GT is the true value map of the infection area segmentation; M_i^{up} represents the up-sampling of the picture M_i to the same size as the true value map GT.

In addition, we use L_e (see Equation 2) to monitor the output of the EAM. Therefore, the total loss function is calculated from the following equation:

$$L_{total} = \sum_{i=2}^{i=5} L_i + L_e. \tag{11}$$

2.6 | Evaluation metrics

We use three popular metrics to evaluate our model: the Dice similarity coefficient, sensitivity (Sen), and specificity (Spec). We also introduce three golden metrics for object detection: structure measure (S_α), enhance-alignment measure (E_φ), and mean absolute error (MAE). During the evaluation, we take M_2 upsample as the final prediction map M_f and compare it with the segmentation truth map M_{GT}.

Dice similarity coefficient ($Dice$) is used to calculate the similarity of two samples. For the best segmentation result, it is 1 and the worst is 0. It can be calculated as follows:

$$Dice = \frac{2TP}{FP + 2TP + FN}, \tag{12}$$

where TP represents the area that is predicted to be infected and actually infected; TN means the area that is predicted to be the background and it is actually the background; FP indicates the part that it is predicted to be an infected area but is in fact the uninfected area; FN stands for the part that is predicted to be the background, which is actually the infected area.

Sensitivity (Sen) is the proportion of the predicted infected area to all infected areas. Sen is calculated as follows:

$$Sen = \frac{TP}{TP + FN}. \tag{13}$$

Specificity ($Spec$) is the proportion of the recognized background versus the total background. It can be calculated as follows:

$$Spec = \frac{TN}{FP + TN}. \tag{14}$$

Structure measure (S_α) can measure the structural similarity between the prediction map (M_f), and the truth map (M_{GT}), and be calculated from Equation (15):

$$S_\alpha = (1 - \alpha) \times S_o + \alpha \times S_r, \tag{15}$$

where α refers to the balance coefficient, which is set to 0.5 by default; S_o represents the object level similarity, and S_r denotes the regional level similarity.

Enhance-alignment measure (E_φ) measures the local and global similarity between M_f and M_f at the same time, and it can be expressed as follows:

$$E_\varphi = \frac{1}{w \times h} \sum_x^w \sum_y^h \emptyset\left(M_f(x,y), M_{GT}(x,y)\right), \tag{16}$$

where w and h are the width and height of the truth map M_{GT}, respectively; (x, y) means the pixel coordinates, and $\emptyset$ represents the enhanced alignment matrix. We report the mean value of all E_φ in the experiment, denoted as $E_\emptyset^{mean}$.

MAE indicates the average of the absolute errors between the predicted value and the observed value. We employ MAE to measure the error between M_f and M_f at the pixel level, which is defined as

$$MAE = \frac{1}{w \times h} \sum_x^w \sum_y^h \left|M_f(x,y), -(x,y)\right|. \tag{17}$$

3 | RESULTS

Our model is implemented under the PyTorch framework on the Ubuntu V20.04 distribution. The hardware environment includes CPU, Intel i7-4790; GPU, NVIDIA GTX TITAN X (12G); RAM, 32G.

3.1 | Experimental details

3.1.1 | Dataset

The dataset we used consists of 100 labeled CT slices from the COVID-19 CT segmentation dataset.[47] All the CT images were from more than 40 COVID-19 patients and collected by the Italian Society of Medical and Interventional Radiology. A radiologist segmented the CT images using different labels for identifying lung infections. It is the first open-access COVID-19 dataset for lung infection segmentation, but with a small size. Inspired by literature,[21] we divided 100 labeled CT images into train, validation and test datasets, which consists of 45 CT images randomly selected as training samples, 5 CT images for validation, and the remaining 50 images for testing. CT slices do not have a uniform resolution. Before training, we uniformly resize the resolution of all CT slices to 352×352.

3.1.2 | Training parameters

Because these CT images do not have a uniform resolution, we have to resize them to 352×352 before training. We train Inf-Net using a multi-scale strategy.[33] Specifically, we first resample the training images using different scaling ratios, that is, (0.75, 1, 1.25), and then train Inf-Net using the resampled images, which improves the generalization of our model. In addition, the learning rate is set to $1e-4$. The batch size is 2, and the training epoch is 100.

3.2 | Comparative experiment

To evaluate our method comprehensively, we compared it with the state of the arts: U-Net,[25] U-Net++,[22] Attention U-Net,[48] Gated-UNet,[49] Dense-UNet,[54] and Inf-Net.[21] In order to compare the performance of these related networks fairly, all of them are trained with the same network parameter settings. In addition, we evaluate our method on six golden evaluation metrics mentioned earlier: The smaller the average absolute error (MAE) value is, the better the performance is; the larger the other index values is, the better the performance is. The segmentation results of the COVID-19 infection area are listed in Table 1, and the best results of each metrics are marked in bold.

From Table 1, it can be seen that our proposed network (Ref-Net) performs well. Compared with the latest COVID-19 segmentation model Inf-Net, the Dice similarity coefficient of our model has increased by 7.3%; the sensitivity (Sen) has increased by 5.9%, and the other evaluation indicators: *Spec*, S_α, E_ϕ^{mean} have increased by 1.6%, 1.4%, and 8.1%, respectively; meanwhile, the *MAE* has dropped by 2.2%.

The PR (precision-recall) curve of U-Net, U-Net++, Inf-Net, and our Ref-Net is shown in Figure 6a. If the PR curve of one network *X* surrounds the PR curve of another network *Y*, it can be concluded that the performance of network *X* is better than *Y*. Our algorithm in Figure 6a is represented by a red bold line. It can be seen that the PR curve of ours surrounds other networks, so our network is better than others. Figure 6b shows the evaluation result of another evaluation metric: *F*-measure. *F*-Measure weighs the precision rate and the recall rate and averages them to consider them comprehensively, because the precision rate and recall rate sometimes conflict. Obviously, the most commonly used evaluation method is *F*-measure, sometimes. Therefore, we use both PR curve and *F*-measure curve to show the outperformance of our algorithm. From Figure 6b, we can also see that the *F*-measure curve of ours still surrounds other networks, so our network is better than others indeed.

To further evaluate our method, some representative results of the COVID-19 CT image segmentation are shown in Figure 7. From Figure 7, we can see that our method has clearer boundaries and has fewer false negatives and false positives. From Figure 7, it is very intuitive to see that our method is better, and the segmentation results are more accurate.

We also found that our algorithm has good portability and is suitable for various current popular networks. To prove this viewpoint, we further compare our network with Inf-Net. We replaced Inf-Net and Ref-Net with other backbone networks on the same dataset, and the COVID-19 CT image segmentation results are shown in Table 2.

In Table 2, to verify the superiority of Res2Net as the backbone network, we replaced the backbone network of ours with VGG16 or ResNet by using the same network parameter settings. We can see that the result of Res2Net as the backbone network is the best. Furthermore, we carried out a comparative experiment of Ref-Net and Inf-Net using different backbone networks. We can see that our method has a significant improvement in *Dice*, E_ϕ^{mean}, and MAE compared to Inf-Net. Therefore, we can conclude that our algorithm is still superior to Inf-Net when using a different backbone network.

3.3 | Ablation experiment

In order to verify the effectiveness of the attention positioning module (APM), we discuss the location of APM in this subsection. On the basis of Ref-Net, we add the attention positioning module to the feature layers f_2, f_3,

TABLE 1 Quantitative results of infection area segmentation

Methods	Backbone	Dice	Sen	Spec	S_α	E_ϕ^{mean}	MAE
U-Net	VGG16	0.439 ± 0.007	0.534 ± 0.006	0.858 ± 0.007	0.622 ± 0.004	0.625 ± 0.002	0.186 ± 0.015
Attention U-Net	VGG16	0.583 ± 0.008	0.637 ± 0.01	0.921 ± 0.003	0.744 ± 0.004	0.739 ± 0.003	0.112 ± 0.01
Gated-UNet	VGG16	0.623 ± 0.005	0.658 ± 0.008	0.926 ± 0.006	0.814 ± 0.003	0.814 ± 0.002	0.102 ± 0.013
Dense-UNet	DenseNet161	0.515 ± 0.006	0.594 ± 0.005	0.840 ± 0.002	0.662 ± 0.001	0.662 ± 0.005	0.184 ± 0.01
U-Net++	VGG16	0.581 ± 0.01	0.672 ± 0.007	0.902 ± 0.003	0.720 ± 0.002	0.720 ± 0.004	0.120 ± 0.015
Inf-Net	Res2Net	0.682 ± 0.008	0.692 ± 0.012	0.943 ± 0.004	0.781 ± 0.002	0.838 ± 0.002	0.082 ± 0.008
Ref-Net (ours)	Res2Net	**0.755 ± 0.005**	**0.751 ± 0.009**	**0.959 ± 0.002**	**0.795 ± 0.001**	**0.919 ± 0.01**	**0.060 ± 0.006**

Abbreviation: MAE, mean absolute error.

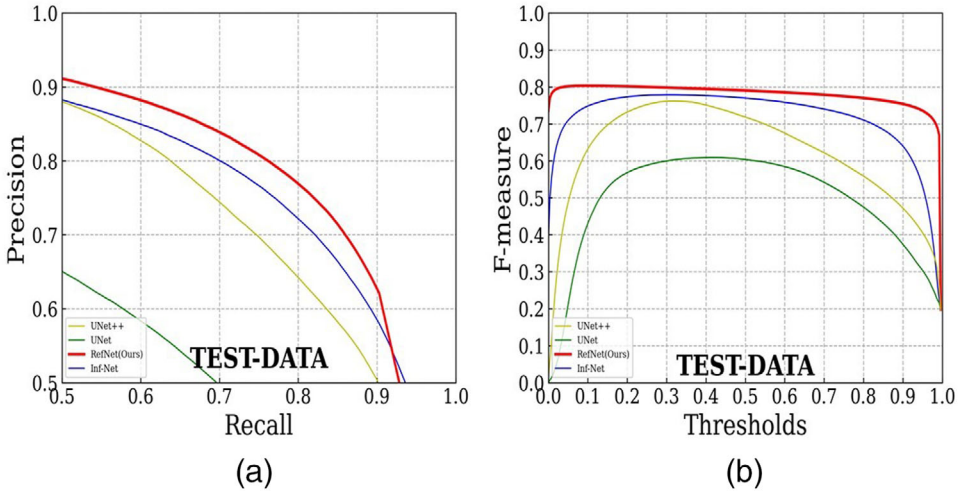

FIGURE 6 Precision–recall curve (a) and *F*-measure (b)

TABLE 2 Compares the performance of Inf-Net with ours (Ref-Net) on different backbone networks

Methods	Backbone	Dice	Sen	Spec	S_α	E_ϕ^{mean}	MAE
Inf-Net	VGG16	0.695 ± 0.006	0.705 ± 0.01	0.930 ± 0.003	0.760 ± 0.003	0.824 ± 0.005	0.075 ± 0.006
Ref-Net	VGG16	0.748 ± 0.004	0.719 ± 0.008	0.961 ± 0.002	0.782 ± 0.002	0.910 ± 0.007	0.059 ± 0.004
Inf-Net	ResNet	0.680 ± 0.007	0.695 ± 0.01	0.940 ± 0.002	0.784 ± 0.003	0.835 ± 0.006	0.073 ± 0.007
Ref-Net	ResNet	0.754 ± 0.005	0.743 ± 0.007	0.960 ± 0.001	0.791 ± 0.001	0.914 ± 0.005	0.059 ± 0.004
Inf-Net	Res2Net	0.682 ± 0.008	0.692 ± 0.012	0.943 ± 0.004	0.781 ± 0.002	0.838 ± 0.002	0.082 ± 0.008
Ref-Net	Res2Net	**0.755 ± 0.005**	**0.751 ± 0.009**	**0.959 ± 0.002**	**0.795 ± 0.001**	**0.919 ± 0.01**	**0.060 ± 0.006**

Abbreviation: MAE, mean absolute error.

f_4, and f_5, respectively, and try to find the best position of the APM.

The working procedures of ablation experiments are as follows (take the addition of the attention positioning module in f_4 as an example): Without changing the position and number of other modules, the attention positioning module after the feature layer f_5 is deleted at first. The feature layer f_5 and its directly generated prediction map are used as the input of the first CEM, which replaces the feature layer and prediction of f_5 after the attention location module in the original model. Subsequently, f_4 is input into the APM, and its accordingly output is regarded as the low-level input of the first context exploration. By following the previous process, attention positioning modules are added to f_2 and f_3. The results are shown in Table 3 (the best results are marked in red). It can be seen from Table 3 that adding the attention positioning module (APM) at a higher level (f_5) has the best performance. This is because that the high level of the CNN is rich in the global positioning semantic information of the object and can locate the target object more comprehensively and accurately.

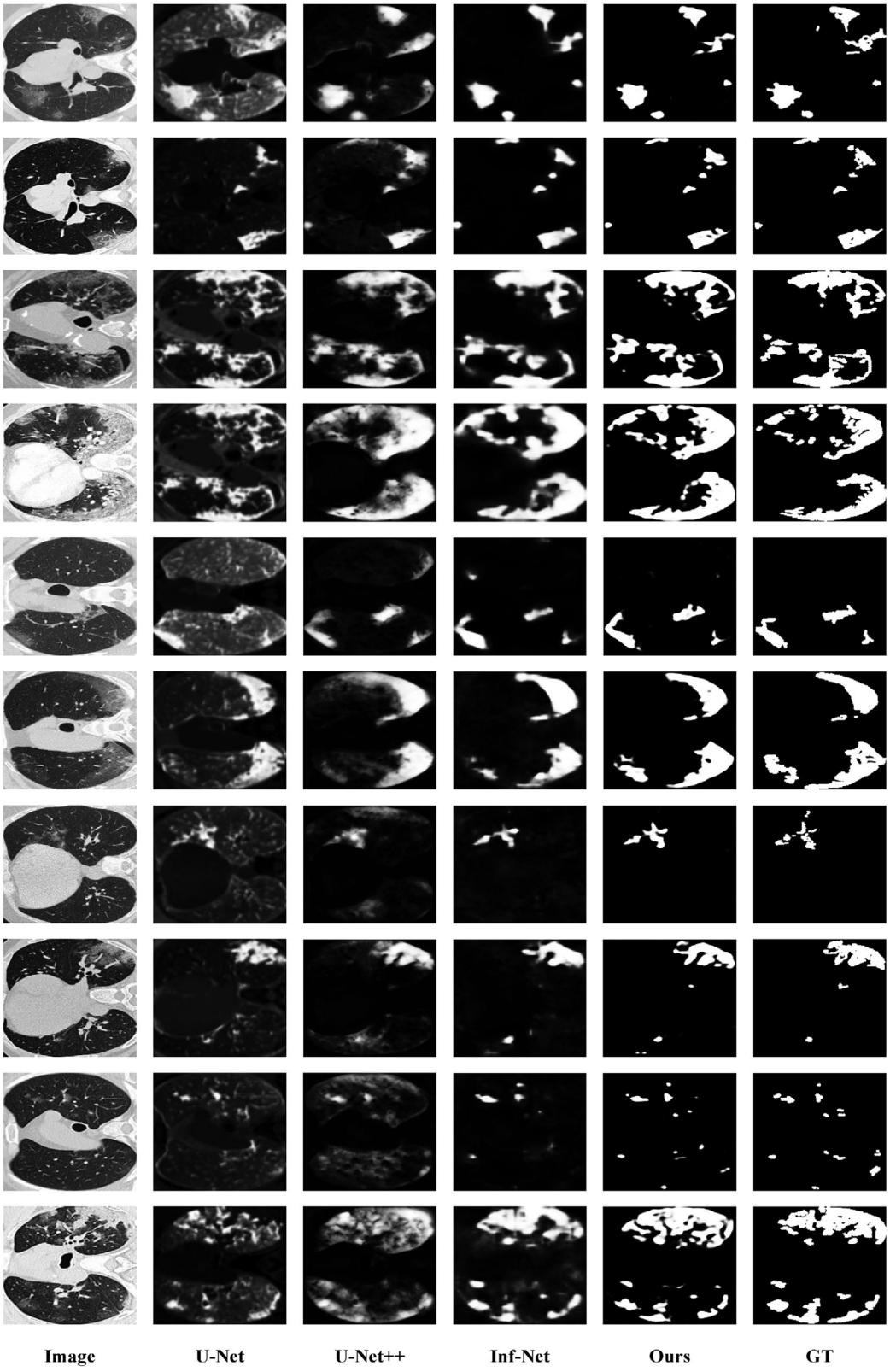

FIGURE 7 Visual comparison of infection area segmentation result

TABLE 3 Ablation experiment of attention position module

Module Setting	Dice	Sen	Spec	S_α	E_ϕ^{mean}	MAE
f_2 + APM	0.747	0.720	0.948	0.788	0.871	0.071
f_3 + APM	0.743	0.725	0.953	0.787	0.913	0.066
f_4 + APM	0.746	0.736	0.951	0.790	0.915	0.068
f_5 + APM	**0.755**	**0.751**	**0.959**	**0.795**	**0.919**	**0.060**

Abbreviation: MAE, mean absolute error.

TABLE 4 Ablation experiment of network model

Module setting	Dice	Sen	Spec	S_α	E_ϕ^{mean}	MAE
(No. 1) Backbone	0.442	0.570	0.825	0.651	0.569	0.207
(No. 2) Backbone + EAM	0.548	0.727	0.765	0.673	0.661	0.231
(No. 3) Backbone + APM	0.551	0.504	0.943	0.679	0.723	0.113
(No. 4) Backbone + CEM	0.626	0.608	0.953	0.736	0.790	0.086
(No. 5) Backbone + EAM + APM	0.605	0.618	0.933	0.711	0.807	0.098
(No. 6) Backbone + EAM + CEM	0.751	0.727	**0.962**	0.782	0.907	**0.058**
(No. 7) Backbone + APM + CEM	0.626	0.592	0.969	0.769	0.794	0.082
(No. 8) Backbone + EAM + APM + CEM	**0.755**	**0.751**	0.959	**0.795**	**0.919**	0.060

Abbreviations: APM, attention position module; CEM, context exploration module; EAM, edge attention module; MAE, mean absolute error.

We also conducted an ablation experiment to ensure that the effectiveness of each module. The experimental results are shown in Table 4.

In Table 4, we take Res2Net as the backbone, whose evaluation metrics are shown in row 1. In order to evaluate the effectiveness of a single module, the EAM, attention positioning module (APM), and CEM were added to the backbone network Res2Net, respectively, whose corresponding results are listed in rows 2–4 of Table 4, respectively. It can be seen that the performance of adding any module could have been improved. That is to say, the EAM, APM, and CEM are all effective actually. Then we arrange and combine the various modules to find the best collocation model. The results are shown in rows 5–8. From Table 4, it can be concluded that the network we designed (no. 8) performs best.

4 | DISCUSSION

CT is considered a low-cost, accurate, and efficient diagnostic tool for screening and diagnosis of COVID-19. It can assess the severity of lung infection. Therefore, the accurate segmentation of infected areas from CT images by automatic segmentation technology can help doctors to quickly screen COVID-19 patients and reduce the workload of doctors.

We designed CT image segmentation network using Res2Net as the backbone network, and EAM, APM, and CEM were used to further improve the segmentation results. According to literature 26, low-level features in shallower convolutional layers preserve spatial information for representing edges, whereas deep convolutional layers preserve semantic information for locating objects. Therefore, we send the low-level features f_2 into the EAM, while feeding the high-level features f_5 into our designed APM. But, the COVID-19 infection area is small and blurred resulting in false negatives and false positives in the segmentation results. Fortunately, context can enhance the ability of feature expression and can eliminate false negatives and false positives to great extent. Therefore, we carefully design a parallel CEM to gradually output accurate segmentation results. Experimental results in Table 1 and Figure 7 both demonstrate the superiority of our method.

From Figure 7, we can easily see that the segmentation result of Inf-Net is better than U-Net and U-Net++, because Inf-Net has own reverse attention. But there are still many ambiguities in the prediction results of Inf-Net; moreover, it does not pay attention to the low-level boundary information that leads to unclear boundaries appeared. By contrary, the segmentation results of our method are clearer than those of Inf-Net. On the one hand, the results of our method remove the ambiguity area in the Inf-Net prediction result. On the other hand, some uninfected areas are segmented (false positives), whereas some infected areas are not detected (false negatives) in the Inf-Net segmentation results. But our algorithm Ref-Net eliminates these false positive regions and false negative regions to the greatest extend, which is undoubtedly contributed to the more accurate final results. In our viewpoint, this is because

there are three CEMs in our model to remove false positives and false negatives altogether.

In future work, we plan to use semi-supervised learning[51–53] to further improve the accuracy of COVID-19 CT image segmentation. In addition, the trust of doctors and patients is required to finally apply deep models to clinical practice. Therefore, the research on the interpretability[54] of deep models is also essential.

5 | CONCLUSION

We propose a segmentation network for segmenting COVID-19-infected regions in CT images. On the public datasets, our network performs very well. Our method can automatically segment infected areas in COVID-19 CT images effectively and efficiently, helping clinicians to screen infected patients, and reducing their burden.

ACKNOWLEDGMENT

This work was supported by the key scientific research project of higher school of Henan Province under Grant no. 21A520022.

CONFLICT OF INTEREST

We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

REFERENCES

1. Wang W, Xu Y, Gao R, et al. Detection of SARS-CoV-2 in different types of clinical specimens. *JAMA*. 2020;323(18):1843-1844.
2. West CP, Montori VM, Sampathkumar P. COVID-19 testing: the threat of false-negative results. *Mayo Clin Proc*. 2020;95(6):1127-1129. Elsevier.
3. Fang Y, Zhang H, Xie J, et al. Sensitivity of chest CT for COVID-19: comparison to RT-PCR. *Radiology*. 2020;296(2):E115-E117.
4. Yang Y, Yang M, Shen C, et al. Evaluating the accuracy of different respiratory specimens in the laboratory diagnosis and monitoring the viral shedding of 2019-nCoV infections. *MedRxiv*. 2020.
5. Wikramaratna P, Paton RS, Ghafari M, et al. Estimating false-negative detection rate of SARS-CoV-2 by RT-PCR. *MedRxiv*. 2020:2020.
6. Guan W, Ni Z, Hu Y, et al. Clinical characteristics of coronavirus disease 2019 in China. *N Engl J Med*. 2020;382(18):1708-1720.
7. Ye Z, Zhang Y, Wang Y, et al. Chest CT manifestations of new coronavirus disease 2019 (COVID-19): a pictorial review. *Eur Radiol*. 2020;30(8):4381-4389.
8. Fang Y, Zhang H, Xie J, et al. Sensitivity of chest CT for COVID-19: comparison to RTPCR. *Radiology*. 2020;296(2):E115-E117.
9. Ai T, Yang Z, Hou H, et al. Correlation of chest CT and RT-PCR testing for coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases. *Radiology*. 2020;296(2):E32-E40.
10. Ng MY, Lee EYP, Yang J, et al. Imaging profile of the COVID-19 infection: radiologic findings and literature review. *Radiol Cardiothorac Imaging*. 2020;2(1):e200034.
11. Keshani M, Azimifar Z, Tajeripour F, et al. Lung nodule segmentation and recognition using SVM classifier and active contour modeling: a complete intelligent system. *Comput Biol Med*. 2013;43(4):287-300.
12. Wang S, Zhou M, Liu Z, et al. Central focused convolutional neural networks: developing a data-driven model for lung nodule segmentation. *Med Image Anal*. 2017;40:172-183.
13. Jiang J, Hu YC, Liu CJ, et al. Multiple resolution residually connected feature streams for automatic lung tumor segmentation from CT images. *IEEE Trans Med Imaging*. 2018;38(1):134-144.
14. Tian D, Han Y, Wang B, et al. Review of object instance segmentation based on deep learning. *J Electron Imaging*. 2021;31(4):041205.
15. Wang Y, Zhang Y, Liu Y, et al. Does non-COVID-19 lung lesion help? Investigating transferability in COVID-19 CT image segmentation. *Comput Methods Programs Biomed*. 2021;202:106004.
16. Müller D, Rey IS, Kramer F. Automated chest CT image segmentation of COVID-19 lung infection based on 3d U-Net. arXiv preprint arXiv:2007.04774. 2020.
17. Abd Elaziz M, Ewees AA, Yousri D, et al. An improved marine predators algorithm with fuzzy entropy for multi-level thresholding: real world example of COVID-19 CT image segmentation. *IEEE Access*. 2020;8:125306-125330.
18. Yan Q, et al. COVID-19 chest CT image segmentation – a deep convolutional neural network solution. arXiv preprint arXiv:2004.10987. 2020.
19. Wang S, Kang B, Ma J, et al. A deep learning algorithm using CT images to screen for corona virus disease (COVID-19). *Eur Radiol*. 2021;31(8):6096-6104.
20. Chen J, Wu L, Zhang J, et al. Deep learning-based model for detecting 2019 novel coronavirus pneumonia on high-resolution computed tomography. *Sci Rep*. 2020;10(1):1-11.
21. Zhou Z, Rahman Siddiquee MM, Tajbakhsh N, et al. Unet++: a nested U-Net architecture for medical image segmentation. *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer; 2018:3-11.
22. Fan DP, Zhou T, Ji GP, et al. Inf-Net: automatic covid-19 lung infection segmentation from CT images. *IEEE Trans Med Imaging*. 2020;39(8):2626-2637.
23. Saood A, Hatem I. COVID-19 lung CT image segmentation using deep learning methods: U-Net versus SegNet. *BMC Med Imaging*. 2021;21(1):1-10.
24. Badrinarayanan V, Kendall A, Cipolla R. SegNet: a deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans Pattern Anal Mach Intell*. 2017;39(12):2481-2495.
25. Ronneberger O, Fischer P, Brox T. U-Net: convolutional networks for biomedical image segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer; 2015:234-241.
26. Zhao T, Wu X. Pyramid feature attention network for saliency detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019:3085-3094.
27. Hou Q, Cheng MM, Hu X, et al. Deeply supervised salient object detection with short connections. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017:3203-3212.
28. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016:770-778.
29. Xie S, Girshick R, Dollár P, et al. Aggregated residual transformations for deep neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017:1492-1500.
30. Gao SH, Cheng MM, Zhao K, et al. Res2net: a new multi-scale backbone architecture. *IEEE Trans Pattern Anal Mach Intell*. 2019;43(2):652-662.
31. Zhao JX, Liu JJ, Fan DP, et al. EGNet: edge guidance network for salient object detection. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019:8779-8788.

32. Wu Z, Su L, Huang Q. Stacked cross refinement network for edge-aware salient object detection. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019:7264-7273.
33. Zhang Z, Fu H, Dai H, et al. ET-Net: a generic edge-attention guidance network for medical image segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer; 2019:442-450.
34. Hu J, Shen L, Sun G. Squeeze-and-excitation networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018:7132-7141.
35. Wang Q, Wu B, Zhu P, et al. ECA-Net: efficient channel attention for deep convolutional neural networks. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE; 2020.
36. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. *Adv Neural Inf Process Syst*. 2017;30:5998-6008.
37. Fu J, Liu J, Tian H, et al. Dual attention network for scene segmentation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019:3146-3154.
38. Mei H, Liu Y, Wei Z, et al. Exploring dense context for salient object detection. *IEEE Trans Circuits Syst Video Technol*. 2021;32(3):1378-1389.
39. Chen LC, Papandreou G, Kokkinos I, et al. Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans Pattern Anal Mach Intell*. 2017;40(4):834-848.
40. Zhang J, Long C, Wang Y, et al. Multi-context and enhanced reconstruction network for single image super resolution. 2020 IEEE International Conference on Multimedia and Expo (ICME). IEEE; 2020:1-6.
41. Ding H, Jiang X, Shuai B, et al. Context contrasted feature and gated multi-scale aggregation for scene segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018:2393-2402.
42. Mei H, Yang X, Wang Y, et al. Don't hit me! glass detection in real-world scenes. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 3687-3696.
43. Zheng Q, Qiao X, Cao Y, et al. Distraction-aware shadow detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019:5167-5176.
44. Mei H, Ji GP, Wei Z, et al. Camouflaged object segmentation with distraction mining. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021:8772-8781.
45. Qin X, Zhang Z, Huang C, et al. Basnet: boundary-aware salient object detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019:7479-7489.
46. Wei J, Shuhui W and Qingming H. F^3Net: fusion, feedback and focus for salient object detection. Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34. No. 07. 2020.
47. Jenssen HB. COVID-19 CT segmentation dataset. 2020. Availableat: https://medicalsegmentation.com/about
48. Oktay O, et al. Attention U-Net: learning where to look for the pancreas. arXiv preprint arXiv:1804.03999. 2018.
49. Schlemper J, Oktay O, Schaap M, et al. Attention gated networks: learning to leverage salient regions in medical images. *Med Image Anal*. 2019;53:197-207.
50. Li X, Chen H, Qi X, et al. H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes. *IEEE Trans Med Imaging*. 2018;37(12):2663-2674.
51. Van Engelen JE, Hoos HH. A survey on semi-supervised learning. *Mach Learn*. 2020;109(2):373-440.
52. Zhai X, Oliver A, Kolesnikov A, et al. S4l: self-supervised semi-supervised learning. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019:1476-1485.
53. Zhou ZH. Semi-supervised learning. *Machine Learning*. Springer; 2021:315-341.
54. Vellido A. The importance of interpretability and visualization in machine learning for applications in medicine and health care. *Neural Comput Appl*. 2020;32(24):18069-18083.

How to cite this article: Liu S, Tang X, Cai T, Zhang Y, Wang C. COVID-19 CT image segmentation based on improved Res2Net. *Med Phys*. 2022;1-13.
https://doi.org/10.1002/mp.15882