COVSeg-NET: A deep convolution neural network for COVID-19 lung CT image segmentation

XiaoQing Zhang¹ | GuangYu Wang² | Shu-Guang Zhao²

¹Taizhou Institute of Science and Technology, Nanjing University of Science and Technology, No.8, Meilan East Road, Taizhou, China
²College of Information Science and Technology, Donghua University, Shanghai, China

Correspondence
XiaoQing Zhang, Taizhou Institute of Science and Technology, Nanjing University of Science and Technology, No.8, Meilan East Road, Taizhou, 225300, China.
Email: zxq_005@163.com

Abstract
COVID-19 is a new type of respiratory infectious disease that poses a serious threat to the survival of human beings all over the world. Using artificial intelligence technology to analyze lung images of COVID-19 patients can achieve rapid and effective detection. This study proposes a COVSeg-NET model that can accurately segment ground glass opaque lesions in COVID-19 lung CT images. The COVSeg-NET model is based on the fully convolutional neural network model structure, which mainly includes convolutional layer, nonlinear unit activation function, maximum pooling layer, batch normalization layer, merge layer, flattening layer, sigmoid layer, and so forth. Through experiments and evaluation results, it can be seen that the dice coefficient, sensitivity, and specificity of the COVSeg-NET model are 0.561, 0.447, and 0.996 respectively, which are more advanced than other deep learning methods. The COVSeg-NET model can use a smaller training set and shorter test time to obtain better segmentation results.

Keywords
convolution neural network, COVID-19, image segmentation, lung CT image

1 | INTRODUCTION

COVID-19 is a new type of respiratory infectious disease. It broke out for the first time in November 2019 and quickly swept the world in less than a year, posing a serious threat to world development and human survival. It is the most challenging problem facing the world today. The disease is caused by severe acute respiratory syndrome coronavirus –2 (SARS-COV-2), which has caused high incidence rate and mortality worldwide.¹ At present, the main clinical tool for the diagnosis of COVID-19 is reverse transcription polymerase chain reaction (RT-PCR), but the detection reagent is expensive and insufficient in quantity, and it needs specialized medical personnel to use it.² This has also led to many suspicious and asymptomatic patients not being detected in time, resulting in a wider range of infections. In order to alleviate the shortage of RT-PCR kits and improve the accuracy of detection, medical staff have been trying to find other diagnostic methods to assist diagnosis. As a noninvasive imaging method, computed tomography (CT) can display the characteristic lung manifestations related to COVID-19.³,⁴ Most of the COVID-19 cases have similar imaging features, including localized inflammatory infiltration in unilateral or bilateral lungs in the early stage, with patchy, lumpy, segmental, or sub-segmental ground glass opacity (GGO) in the lower pleura, and consolidation in the lung in the late stage.⁵ Therefore, the analysis of lung CT images can be used as an effective means for early screening and diagnosis of COVID-19. However, due to the overlap of lung images of viral pneumonia, it is difficult to judge whether patients are infected with COVID-19 by doctors alone, and the number of professional doctors is also seriously
lacking. Therefore, it is of great significance to assist doctors in complex detection tasks with AI technology.

In order to facilitate the researchers to study the coronavirus, Yang et al.\(^6\) constructed an open-source COVID-19 lung CT image data set, including 349 COVID-19 lung CT images from 216 patients and 463 lung CT images from non-COVID-19 patients. The practicability of these data set has been recognized by a senior radiologist and has been used by many researchers to train their own deep network model. Chaia et al.\(^7\) proposed a deep convolutional neural network (CNN) model named CVDNet. Based on the residual network, the model uses multiple parallel layers with different kernel sizes to detect the local and global features of images. The trained model can be used to distinguish lung images of normal people, patients with COVID-19, and other patients with viral pneumonia, and the accuracy rate reaches 96.69\%.

Wang et al.\(^8\) proposed COVID-Net, which is a deep CNN (volatile neural networks) network that can distinguish COVID-19 patients from chest X-ray images. The overall accuracy of COVID net can reach 83.5\% by classifying the four types of images of COVID-19 infection, viral pneumonia, bacterial pneumonia, and normal. The overall accuracy of classification of COVID-19, normal, and non-COVID pneumonia cases was 92.4\%. Shayan et al.\(^9\) proposed a CNN framework to find the lesion area in the lung image. The experimental results show that the accuracy of this method for lung image segmentation can reach 83.84\%. Hemdan et al.\(^10\) proposed a model named COVIDX to help radiologists automatically detect COVID-19 from chest X-ray images. The framework is based on seven deep-seated architectures, namely, VGG19, InceptionV3, Xception, DenseNet201, MobileNetV2, InceptionResNetV2, and ResNetV2. After model training, the author tested on 50 X-ray images. In this test set, 25 cases were infected with COVID-19 and 25 cases were not. Finally, the author got the conclusion: VGG19 and DenseNet models have similar performance on automatic detection of COVID-19, with F1 values of 0.91 and 0.89, respectively, while the classification performance of InceptionV3 model is poor, the F1 values of COVID-19 and normal conditions are 0.00 and 0.67, respectively. Considering that the use of deep learning technology for image segmentation often requires a large number of manually labeled data, supervised training, but it is often difficult to obtain these data in the early stage of disease outbreak. Mohamed et al.\(^11\) proposed a new FSS (few-shot segmentation) dual-path deep learning semi-supervised architecture fss-2019-ncov, which consists of a feature encoder module based on ResNet34, a context enrichment module composed of smoothed atrous convolution block and multiscale pyramid pooling block and a feature decoder module. It overcomes the limitation of lacking a large number of COVID-19 CT images. At the same time, the author proves that the model has good effect through experiments. In the segmentation task of GGO area of lung, the sensitivity of the model is 0.768, and the specificity is 0.980. At the same time, Zheng et al.\(^12\) developed a COVID-19 detection software system based on weak supervised deep learning. The system first uses U-Net to segment lung regions, and then inputs the segmented three-dimensional lung regions into three-dimensional deep neural network to predict the probability of COVID-19 infection. Four hundred and ninety-nine CT images were used for training and 131 CT images were used for testing. The results show that the accuracy of the algorithm is 0.901, the positive prediction accuracy of the COVID-19 is 0.840, and the negative prediction accuracy is 0.982. Adel et al.\(^13\) designed and evaluated a method for automatic segmentation and measurement of COVID-19 pulmonary infection using chest CT image, and compared the method with GraphCut, medical image segmentation (MIS) and Watershed segmentation, and concluded that this end-to-end learning method has better efficiency and flexibility in CT image segmentation. The accuracy, sensitivity, F-measure, precision, MCC, dice, jacquard, and specificity were 0.98, 0.73, 0.71, 0.73, 0.71, 0.57, and 0.99, respectively.

In this research, we have absorbed the experience of our predecessors in processing medical images, combined with the latest advanced methods, and proposed a COVSeg-NET model for segmentation of COVID-19 lung lesions. The COVSeg-NET model can help radiologists to quickly find the infected area of the COVID-19 lung CT image, and make a preliminary judgment on whether the patient has COVID-19, so as to carry out targeted treatment of the diseased area. This model can greatly improve the efficiency of diagnosis and treatment. The specific contents of this article are as follows: 1. Data preparation: This study collected CT images and corresponding masks of 929 COVID-19 patients. 2. The design and research of the network model: This study improves the traditional model,\(^12\) and obtains a COVSeg-NET model that is more suitable for lung lesion segmentation. 3. Evaluation of the results of lung CT image segmentation for COVID-19: Experiments show that the COVSeg-NET model has better results in lung CT image segmentation.

## 2 Convolution Neural Network and Related Concepts

According to the characteristics of COVID-19 lung CT images, COVSeg-NET model was designed after analysis
and research. This model is a full convolution neural network model. The left half of the network model is a series of down-sampling operations composed of convolution, ReLU/ELU, and pooling. After each downsampling, the number of feature map is doubled. The right part of the network model doubles the size of the feature map by deconvolution and reduces its number by half. Then, the feature map on the right merges with the feature map of the left symmetric compression path, and outputs the image segmentation region. The error between the segmented region obtained by the model and the real segmentation region is back-propagation to train the network model.

2.1 Convolutional neural network

CNN is a structure which can extract more and more complex features layer by layer. The image is input into the network model as a matrix, and two-dimensional filter is needed for two-dimensional convolution. Given an image \( x_{ij}, 1 \leq i \leq M, 1 \leq j \leq N \) and a filter \( f_{ij}, 1 \leq i \leq m, 1 \leq j \leq n \), the output of convolution is shown in formula (1):

\[
y_q = \sum_{u=1}^{m} \sum_{v=1}^{n} f_{uv} x_{i-u+1,j-v+1}
\]

The input of the \( i \) the neuron in the \( l \) layer is \( a_{i}^{l} = f \left( \sum_{j=1}^{m} w_{ij}^{(l)} a_{i-j+m}^{(l-1)} + b^{(l)} \right) \) or \( a^{(l)} = f(w^{(l)} \otimes a^{(l-1)} + b^{(l)}) \), where \( \otimes \) represents convolution operation and \( w^{(l)} \) shares the weights of neurons in layer \( l \). You can use \( k \) different filters to get \( k \) sets of outputs. The \( k \)-th feature mapping of layer \( l \) is shown in formula (2):

\[
X^{(l,k)} = f \left( \sum_{p=1}^{n_{l-1}} \left( w^{(l,k,p)} \otimes X^{(l-1,p)} + b^{(l,k)} \right) \right)
\]

Where \( w^{(l,k,p)} \) is a filter for mapping the \( p \)-th eigenvector of layer \( l-1 \) to the \( k \)-th feature map of layer \( l \). Each set of feature maps in the layer uses \( n_{l-1} \) filters and a bias \( b \).

2.2 Back propagation

Back propagation (BP) is the abbreviation of “error back propagation.” It is a common method used to train artificial neural networks in combination with optimization methods (such as gradient descent method). The gradient of loss function is calculated by all weights in the network, and the gradient is fed back to the optimization method to update the weights to minimize the loss function. For a sample \((x^{(i)}, y^{(i)})\), \( 1 \leq i \leq N \), the objective function is shown in formula (3):

\[
J(W, b) = \sum_{i=1}^{N} L(y^{(i)} f(x^{(i)} | W, b)) + \frac{1}{2} \| W \|_{F}^{2}
\]

\[
= \sum_{i=1}^{N} J(W, b; x^{(i)}, y^{(i)}) + \frac{1}{2} \| W \|_{F}^{2}
\]

where \( f(x | w, b) \) is the output of feedforward neural network, \( W \) and \( b \) contain the weight matrix and bias vector of each layer, \( \| W \|_{F}^{2} = \sum_{l=1}^{L} \sum_{j=1}^{n_{l}} \sum_{i=1}^{n_{l-1}} W_{ij}^{(l)} \). Moreover, using the gradient descent method, the updating parameters are shown in formula 4:

\[
W^{(l)} = W^{(l)} - \alpha \frac{\partial J(W, b)}{\partial W^{(l)}} = W^{(l)} - \alpha \sum_{i=1}^{N} \left( \frac{\partial J(W, b; x^{(i)}, y^{(i)})}{\partial W^{(l)}} \right) - \lambda W
\]

\[
b^{(l)} = b^{(l)} - \alpha \frac{\partial J(W, b)}{\partial b^{(l)}} = b^{(l)} - \alpha \sum_{i=1}^{N} \left( \frac{\partial J(W, b; x^{(i)}, y^{(i)})}{\partial b^{(l)}} \right)
\]

where \( \alpha \) is the update rate of the parameters.

2.3 Learning rate

As an important super parameter in deep learning, learning rate determines whether the objective function converges to the local minimum value and the speed of convergence to the minimum value. The appropriate learning rate can make the objective function converge to the local minimum value in the appropriate time. In the process of gradient descent, if the learning rate is too large, the model is not easy to converge. If it is too small, the convergence speed will be too slow, as shown in Figure 1. The updating formula of model weight is shown in formula (5).

\[
w \leftarrow w - \eta \frac{\partial E}{\partial w}
\]

\[
\eta_{opt} = \left( \frac{\partial^2 E}{\partial w^2} \right)^{-\frac{1}{2}}
\]

where \( w \) is the weight matrix, \( \eta \) is the learning rate, \( \eta_{opt} \) is the optimal learning rate, and \( E \) is the objective function gradient.

There are three methods to calculate the gradient descent and its related variables:
First, all samples are used to calculate the gradient, but the disadvantage of this method is that the calculation speed is slow. For the problem of large amount of data, the computer memory is often insufficient:

$$w_{t+1} = w_t - \eta \nabla_w L(w) \quad (6)$$

Second, a single sample is randomly selected to calculate the gradient. The disadvantage of this method is that the variance is large, and the loss function fluctuates greatly:

$$w_{t+1} = w_t - \eta \nabla_w L(w, x_i, y_i) \quad (7)$$

Third, a random Mini-batch (a small batch of samples composed of N data) is randomly selected to calculate the gradient and to update the parameters (Mini-Batch SGD):

$$w_{t+1} = w_t - \frac{1}{N} \eta \sum_{i=1}^{N} \nabla_w L(w, x_i, y_i) \quad (8)$$

where $L$ is the loss function, $w$ is the weight matrix, and $\eta$ is the learning rate.

### 2.4 Activation function

It is a nonlinear function, which makes the network model have the ability of nonlinear modeling. Otherwise, neural network can only realize linear mapping and cannot deal with some complex problems. At present, the commonly used activation functions mainly include ReLU, ELU, RMSprop, Sigmoid, and so forth, as shown in Figure 2.

### 2.5 Gradient disappearance

When sigmoid function is used as activation function, the error will gradually decrease in the backpropagation through each layer; for deep networks, gradient will even disappear, which is called gradient disappearance.

As shown in Figure 3, the gradient of $b_1$ is calculated by BP algorithm:

$$\frac{\partial C}{\partial b_1} = \sigma(b_1) w_2 \sigma(b_2) w_3 \sigma(b_3) w_4 \sigma(b_4) \frac{\partial C}{\partial b_4} \quad (9)$$

The gradient of sigmoid activation function is shown in formula (10):

$$\sigma(x) = \sigma(x) (1 - \sigma(x)) \in \left(0, \frac{1}{4}\right) \quad (10)$$

Therefore:

$$\frac{\partial C}{\partial b_1} \leq \left(\frac{1}{4}\right)^4 w_2 w_3 w_4 \frac{\partial C}{\partial b_4} \quad (11)$$

In the above formula, $x$ is the input sample and $w$ is the weight; in this case, a rectifier function or similar activation function can be used. That is, when the derivative of the activation function is 1, the error can be passed normally to improve the training speed.

### 2.6 Fully convolutional neural network

Fully convolutional neural network (FCN) is different from traditional convolutional neural networks. FCN uses a convolutional layer to replace the fully connected layer, so that the CNN network originally used for classification can be converted into a segmentation network that can generate a spatial heat map. At the same time, by adding a transposed convolutional layer to the network, the network can perform intensive inference and learn the semantic label of each pixel in the image. The structure of FCN is shown in Figure 4.
2.7 | Feature fusion

Feature fusion can add contextual information to the full convolutional neural network architecture. Its specific method is to combine global features extracted from a shallower network layer with local features extracted from a relatively deep layer. There are two common feature fusion methods: early fusion and late fusion. The early fusion method comes from the context block in ParseNet.15 The early fusion is to fuse multiple levels of features, and then train a predictor on the fused features. Late fusion uses the method of jump connection to perform delayed feature fusion for FCN network utilization. By combining feature maps generated by different layers, it is similar to fusing the results after making independent predictions on each layer. The U-Net segmentation model uses the feature fusion achieved by this jump connection method.

3 | IMPROVEMENT AND EXPERIMENTATION OF LUNG CT IMAGE SEGMENTATION METHOD FOR COVID-19

3.1 | Data set preparation

This study collected CT image data sets from Kaggle's COVID-19 CT image segmentation (imaging performance of lung axial section) competition. Two annotated CT datasets are employed for model evaluation, publicly published by the Italian Society of Medical and Interventional Radiology. The dataset was kindly provided by medicalsegmentation.com, and radiologists do a lot of work such as lung Mask (Ground-Truth) on some images. The first dataset contains 100 axial CT slices and 100 masks belonging to over 40 patients that are positively confirmed to have COVID-19. The images of first dataset were converted from openly accessible JPG images and the size of each slice was set to 512 × 512 pixels. Additionally, the second dataset contains nine CT volumes consisting of 829 slices and masks. Among them, there were 373 annotated axial CT slices that were positively confirmed as COVID-19. The size of each slice in second dataset was set to 630 × 630 pixels. We selected 406 clearer COVID-19-positive lung CT images from these 929 images for the experiment in this design. All of the 406 lung CT images of COVID-19 we selected have a corresponding four-label mask image. The marked areas in the mask images, respectively, are as follows: 0-“ground glass opacity,” 1-“consolidations,” 2-“lungs other,” 3-“background.” The data format of this dataset is .npy format. Since the images in this format are in the form of a matrix, we convert them into an image format that can be displayed. The steps are as follows: First, we use the Load method in the Numpy library to read the data in the Python3 environment. Then, use the From array method in the Image module provided by PIL to
read the image information from the .npy file, and use the Resize method to set the image size to 128 × 128. Finally, use the Imvisava method in the Matplotlib.Pyplot library to save all images in .jpg format. After obtaining the CT image, we extracted the mask image (gray image) of the lungs, GGO, and consolidations according to the four-channel Mask image that comes with the data set. The processed image and its corresponding Mask map are shown in Figure 5. In this case, increasing the size of the training set (number of samples) will help reduce overfitting and improve the accuracy of image analysis. Since the image category is obtained through the central pixel, this study can obtain new images with unchanged category labels by randomly moving, rotating, scaling, and channel shifting the image, thereby effectively increasing the size of the training image set.

3.2 | Structure optimization and parameter setting of COVSeg-NET model

According to the characteristics of COVID-19 lung CT image, this study designed a segmentation model for segmenting the GGO area of COVID-19 lung CT image, namely, the COVSeg-NET model. The COVSeg-NET model is similar to the U-Net model. The model has a total of 216 layers and a total of 1,865,676 parameters. The convolutional layer uses a 3 × 3 filter and uses ELU as the activation function. After the convolutional layer, a pooling layer (sub-sampling layer) is also added to reduce the dimensionality of features, thereby avoiding over-fitting. In this design, max pooling layer is used, the main reason for using max pooling layer is that we need to accurately segment the lesion area when performing segmentation operations. Using max pooling layer can extract feature textures well, so that the segmentation results can save the edges as much as possible. The max pooling layer (Maxpool) uses a 3 × 3 filter, the random inactivation layer (Dropout) uses a parameter of 0.5, and the upsampling layer uses a 2 × 2 filter. The COVSeg-NET model combines convolutional layer 4 and convolutional layer 6, convolutional layer 3 and convolutional layer 7, convolutional layer 2 and convolutional layer 8, convolutional layer 1 and convolutional layer 9 for feature fusion to achieve Flattening of structure. The advantage of this is that the model can maximize the contextual information in the image. Finally, the convolutional layer is connected as the output, and then the Sigmoid function is used to predict the segmentation result. The COVSeg-NET model includes four downsampling and four upsampling. Among them, downsampling can increase the robustness to some small disturbances of the input image, such as image translation, rotation, and so forth, thereby reducing the risk of overfitting, reducing the amount of calculation, and increasing the size of the receptive field. The biggest function of upsampling is to restore and decode the abstract features to the size of the original image, and finally get the segmentation result. The structure of the COVSeg-NET model is shown in Figure 6.

The main building block of the COVSeg-NET model is the convolutional layer, which stacks multiple layers on top of each other to form a feature hierarchy. Each layer can be understood as extracting the features of the previous layer into the hierarchical structure connected to it. The size of the receiving field corresponds to the size of the filter, and the value of each row of the filter is the weight value of the connection between the neurons in this layer and the neurons in the upper layer. The feature map is a corresponding topological arrangement map obtained by a filter in a sliding window manner, using the same specific local spatial nonlinear feature extractor (whose parameters are learned), and moving in each spatial neighborhood of the input plane. Practice shows that the learned filter can be regarded as an edge detector, and the filter can be adjusted to different spatial frequencies, proportions, and directions to perform statistical calculations on the training data. The shallow structure of the network model can capture some simple features of the image, while the deep structure is larger because of

![Figure 5](image_url) COVID-19 lung CT image and its corresponding Mask image. From left to right are the lung CT image, the mask of the four labels corresponding to the lung CT image, the mask of the ground glass opacity area of the lung, the mask of the lung consolidation area, and the mask of the entire lung area.
the perception. Moreover, there are more convolution operations, which can obtain more abstract features in the image. Receptive fields of different sizes have different sensitivity to target objects of different sizes. Large receptive fields are easy to identify large objects, but it is difficult to obtain small objects and edge information in the image. Small receptive fields are needed to help identify such difficult-to-obtain information.

The size of COVID-19 lung CT image training set will affect the training effect of COVSeg-NET model. Because there are only 929 CT images of COVID-19 collected, and many of them have incomplete lung information, so they are not suitable for training. After screening, we selected 406 of them for model training and testing. In order to increase the number of training images, we use the ImageDataGenerator for data augmentation in Keras to flip x, y, randomly cut and scale the existing CT images to achieve data augmentation. During the training of the COVSeg-NET model, the network parameters will be updated after each batch of data training. The advantage of batch training is that the amount of data input to the neural network is small each time, and it can be trained under the condition of limited memory. At the same time, after each Epoch runs all batches, the order of the data can be randomly shuffled and the next Epoch can be performed, which can prevent the neural network from falling into the Local Minima due to the repeated sequence overtraining some batches. The learning rate of the model (\( \eta \)) determines the amount of change in a weight update of the model. The smaller the learning rate, the slower the weight update will be, and the trajectory space will be smoother. The higher the learning rate, the faster the weight update, but it may lead to unstable network weight changes. Therefore, in order to update the weights as soon as possible while maintaining good stability of the network, it is usually necessary to introduce a momentum term:

\[
\Delta \omega(n) = -\eta \frac{\partial E(\omega)}{\partial \omega(n)} + \alpha \Delta \omega(n-1), n = 1, 2, ...
\]

In the formula, the first term on the right side of the equal sign is the correction amount of the backpropagation algorithm, and the second term is the momentum term. Among them, \( \alpha \) represents the momentum factor, which is generally valued at (0, 1). \( \alpha \) controls
the degree of influence of the previous weight update on this weight update. After testing, the learning rate and momentum factor of the COVSeg-NET model are set to 0.045 and 0.9. The optimization and loss algorithms of the COVSeg-NET model mainly use RMSprop and Adam. Among them, RMSprop depends on the global learning rate, which can be applied to non-stationary targets, and Adam is controlled by bias. The learning rate of each iteration has a certain range, which makes the parameters relatively stable. The batch size setting depends on the size of the memory, here it is set to 64. The KerasClassifier class builder takes default parameters and passes them (Epoch and Batch Size) to the model.fit calling function.

In order to prevent overfitting, the COVSeg-NET model is trained with 120 steps. Overfitting is a common problem in deep learning. Since the training sample is only a small part of the real data and cannot truly reflect the distribution of the data, the minimization of empirical risk can easily lead to overfitting. Overfitting is mainly caused by the lack of training data. The method to solve over-fitting is generally to add a parameter regularization term on the basis of minimizing the empirical risk. This item can reduce the parameter space:

\[
\theta^* = \arg\min_{\theta} R(\theta) + \lambda \| \theta \|_2^2
\]

\[
= \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L(y^{(i)}, f(x^{(i)}, \theta)) + \lambda \| \theta \|_2^2, \quad (13)
\]

Among them, \( \| \theta \|_2 \) is the regularization term of the \( L_2 \) norm, and \( \lambda \) is used to control the intensity of regularization. Although in principle, as the model training progresses, the training error of the neural network model based on the training data set should gradually decrease. However, in practice, when the model is complicated to a certain degree, this error will increase with the increase of the complexity of the model, which leads to the phenomenon of over-fitting. Different researchers have different understandings (definitions) of overfitting. For example: ① The error rate of the training set is smaller than that of the test set. ② The error rate of the training set is declining, but the error rate of the test set is beginning to rise. For the first type of overfitting, as long as the number of iterations increases, the error rate of the model based on the test set decreases, and the accuracy of the training set and test set increases. The accuracy of the test set is lower than that of the training set. It can be tolerated, because the error generated by the fitting may exist. The second type of overfitting should be bad in any situation. The over-fitting problem can generally be alleviated by using methods such as early stop, data amplification, regularization, and random dropout. By optimizing the structure of the network model, and obtaining and using more raw data for model training, the problem of model overfitting can be better overcome. Early stopping is the gradient descent training process. Due to the possibility of overfitting, the validation set is used to test whether the parameters of each iteration are the best for the validation set. If the error rate of the validation set test stops decreasing, no iterations will be made. Shown in Figure 7.

The parameter settings for training the neural network model of COVID-19 CT image segmentation are shown in Table 1.

The COVSeg-NET model uses the batch normalization (BN) layer in the convolution process. The BN layer can force the increasingly biased data distribution in model training back to a more standard distribution through certain standardized means, so that the activation input value can fall in the area where the nonlinear function is more sensitive to the input, and the output of the network will not Big. The advantage of using the BN layer is that it can speed up the training and convergence of the network, control the gradient explosion to prevent the gradient from disappearing and prevent overfitting, which can replace the dropout layer to a certain extent. At the same time, the Inception structure is introduced in the COVSeg-NET model. This structure stacks 1×1, 3×3, 5×5 convolutional layers and 3×3 pooling layers together, which increases the width of the network on the one hand, and increases the adaptability of the network to scales on the other hand. The architecture of COVSeg-NET model is shown in Tables 2 and 3. Table 4 describes the relevant parameters of the COVSeg-NET model, including the number of pixels of

![Figure 7](image-url)
the input image of the model, the depth of each convolutional layer in the model, the number of filters in each pooling layer in the model, and the upsampling layer of the model. Table 4 describes the number of pixels of the input image of the COVSeg-NET model structure, the depth of each convolutional layer, the number of filters in each pooling layer, the depth of the upsampling layer, the size of the output feature map, the frame used for model training and other parameter settings.

4 | EVALUATION AND DISCUSSION OF EXPERIMENTAL RESULTS

The COVID-19 lung CT image segmentation experiments are all completed in the HP notebook with the following configuration:

1. Hard Disk: 16GB RAM, 1TB + 256GB
2. Graphics processor: NVIDIA Geforce GTX 1070 8G
3. Central processing unit: Intel Core i7-6700HQ processor
4. Operating system: Ubuntu 16.04 operating system
5. Software: Pycharm 2.7.13, Theano 0.8.2 + Keras 1.2.0, GPU_device, Cuda 8.0, OpenCV 3.4.1, Anaconda 2, Numpy 1.14.0

Open source frameworks for deep learning include TensorFlow, Caffe, Keras, CNTK, Torch, MXNet, Leaf, Theano, DeepLearning4J, Lasagne, Neon, and so forth. Companies such as Google, Microsoft, and Facebook have participated in the development of deep learning frameworks. Keras is a deep learning framework based on TensorFlow and Theano. It makes the implementation process faster and easier by re-encapsulating TensorFlow or Theano, so you do not need to pay attention to too many low-level details. As shown in Figure 8.

Since the COVID-19 lung CT image segmentation model needs to be used to assist diagnosis and treatment, it is necessary to ensure the fidelity of the image segmentation contour to the shape of the real lesion, otherwise any over-recognition (false alarm, false positive) or under-recognition (Missing alarms and false negatives) will threaten the lives of patients and others. MIS has a strong local deviation, although it does not necessarily occupy a large volume, it will cause a large difference in shape. When judging whether a segmentation algorithm performs well, we usually compare the segmentation

**TABLE 1** Parameter settings when training the COVSeg-NET model

| Parameter name          | Parameter value |
|-------------------------|-----------------|
| Learning rate (η)       | 0.045           |
| Momentum factor (α)     | 0.9             |
| Batch size              | 64              |
| Number of epochs        | 120             |
| Optimization and loss function | RMSprop and Adam |

**TABLE 2** The architecture of the COVSeg-NET model, including the name of the layer, the functions used, and the corresponding parameters

| Name               | Method     | Parameter       |
|--------------------|------------|-----------------|
| Convolution1       | Inception  | 32              |
| Pool1              | Nconvolution | 32 × 3 × 3    |
| Pool1              | Dropout    | 0.5             |
| Convolution2       | Inception  | 64              |
| Pool2              | Nconvolution | 64 × 3 × 3    |
| Pool2              | Dropout    | 0.5             |
| Convolution3       | Inception  | 128             |
| Pool3              | Nconvolution | 128 × 3 × 3  |
| Pool3              | Dropout    | 0.5             |
| Convolution4       | Inception  | 256             |
| Pool4              | Nconvolution | 256 × 3 × 3  |
| Pool4              | Dropout    | 0.5             |
| Convolution5       | Inception  | 512             |
| After_conv4        | Rblock     | 1256            |
| Up6                | Merge      | 2 × 2           |
| Convolution6       | Inception  | 256             |
| Convolution6       | Dropout    | 0.5             |
| After_conv3        | Rblock     | 1128            |
| Up7                | Merge      | 2 × 2           |
| Convolution7       | Inception  | 128             |
| Convolution7       | Dropout    | 0.5             |
| After_conv2        | Rblock     | 1,64            |
| Up8                | Merge      | 2 × 2           |
| Convolution8       | Dropout    | 0.5             |
| After_conv1        | Rblock     | 1,32            |
| Up9                | Merge      | 2 × 2           |
| Convolution9       | Inception  | 32              |
| Convolution9       | Dropout    | 0.5             |
| Convolution10      | Convolution | 1 × 1 × 1      |
| output             | Flatten, Dense | Sigmoid      |
| After_conv4        | Rblock     | 1256            |
| Up6                | Merge      | 2 × 2           |

ZHANG ET AL.
results generated by the algorithm with the expert's segmentation standards. When the automatic segmentation algorithm and the expert manual segmentation produces similar results, the automatic segmentation algorithm is considered to be an acceptable alternative to the expert method. In addition, a truly good automatic MIS algorithm requires less time and has better accuracy than interactive expert segmentation.

After the COVSeg-NET model is trained, we put the test set data into the model for testing, and the model will return an automatically predicted mask map of the GGO area of the lungs. Then, according to the predicted results, the corresponding lung CT images were marked with the GGO area of the lungs (circled in red font), and some of the predicted results were obtained. As shown in Figure 9.

We have selected five test results in Figure 9. From Figure 9, we can see that the prediction results of the selected five lung CT images are very good, and the COVSeg-NET model can basically mark the approximate position and basic shape of the GGO in the lung area. But there are some CT images whose prediction results are not very good, as shown in Figure 10.

Figure 10 shows some unsatisfactory segmentation results, in which image 6 and image 7 can predict the general location of the lesion area, but cannot well predict the shape of the lesion. The prediction results of the image 8 and image 9 are very poor, which not only failed to capture the general location of the lesion, but also failed to accurately predict the shape of the lesion. The reason for this situation has a great relationship with the quality of the image because of the presence of GGO and consolidation in the lungs of COVID-19 patients, so the lung consolidation area is easy to interfere with the segmentation of GGO area. In the first four CT images, the GGO area of the lung is relatively easy to distinguish, so it can get a good effect. In the image 8 and 9, the lung consolidation area and GGO area overlap to a large extent. Because the difference between the two regions is difficult to distinguish and the image quality is not high, the prediction results of the model are not good.

Subsequently, we use quantitative analysis to evaluate the performance of the model. Usually by determining a set of statistics to measure the closeness of the segmentation result image to its true value to calculate the degree of overlap. The test images used in this study and the

| TABLE 3 | The structure of inception and Nconvolution |
|----------------|---------------------------------|
| Inception structure | The structure of Nconvolution |
| Convolution2D | Convolution2D |
| ELU | BatchNormalization |
| Convolution2D | ELU |
| BatchNormalization | ELU |
| Max pooling 2D (3 × 3) | |
| Convolution2D | Merge |
| BatchNormalization | ELU |

| TABLE 4 | The parameters of each layer of the COVSeg-NET model |
|----------------|---------------------------------|
| **Name** | COVSeg-NET Model |
| The number of pixels of the input image | 128 × 128 |
| Conv1 layer depth | 32 |
| Number of filters in Pool1 layer | 32 |
| Conv2 layer depth | 64 |
| Number of filters in Pool2 layer | 64 |
| Conv3 layer depth (depth) | 128 |
| Number of filters in Pool3 layer | 128 |
| Conv4 layer depth | 256 |
| Number of filters in Pool4 layer | 256 |
| Conv5 layer depth | 512 |
| After_Conv4 layer depth | 256 |
| After_Conv3 layer depth | 128 |
| After_Conv2 layer depth | 64 |
| After_Conv1 layer depth | 32 |
| The final output feature map size | 128 × 128 |
| Framework used for model training | Theano + Keras |
images manually segmented by experts are all binary mask images, with white foreground and black background. Overlap the “test” image (the mask of the test image predicted by the COVSeg-NET model) of the COVID-19 lung CT and its corresponding “Ground Truth” (the mask manually segmented by an expert). When the test image perfectly matches Ground Truth, the image is determined to overlap most completely. Quantitative analysis by using four evaluation indicators TP, TN, FP, and FN: The pixels that overlap the foreground of the Mask image of the test image and the foreground of the Mask image of Ground Truth will be considered as “true positives” (TP) because they are correctly marked as foreground. Pixels that overlap the
background of the Mask image of the test image with the background of the Mask image of Ground Truth are considered “true negatives” (TN) because they are correctly labeled as the background. Pixels that overlap the foreground of the Mask image of the test image with the background of the Mask image of Ground Truth will be considered a “false positive” (FP) because it should be marked as part of the background. Pixels that overlap the background of the Mask image of the test image with the foreground of the Mask image of Ground Truth will be considered a “false negative” (FN) because it is incorrectly marked as part of the background (but should not be). Figure 11 lists the nine predicted mask images and the real mask images in the COVID-19 lung CT test set, as well as the TN, TP, FN, and FP evaluation results obtained by overlapping them.

It can be seen from the results in Figure 11 that in the first five groups of images, there are fewer white parts in the FP region, which means that the foreground of the mask image predicted by COVSeg-NET in the case of better image quality is basically the same as that of the ground truth mask image. At the same time, it can be seen from the FN area in the nine groups of images that the Mask image predicted by the model has many backgrounds that are incorrectly predicted, so the model needs to be improved. Next, we further use six common indicators to evaluate the performance of the COVSeg-NET model, including: Sensitivity, specificity, dice coefficient, pixel accuracy (Accuracy), precision (Precision), and recall Rate (Recall). Among them, the Dice coefficient is the harmonic average of the precision rate and the recall rate, which takes into account both the precision rate and the recall rate. Table 5 is a performance evaluation of the segmentation results of partial images of the COVID-19 lung CT test set, where nine images correspond to the nine predicted Mask images in Figures 9 and 10, respectively. Through these indexes, the modified segmentation model can be evaluated more objectively, and the index values of TP, TN, FP, and FN can be used for calculation. The formula is as follows:

\[
\text{Sensitivity} = \frac{TP}{(TP + FN)}
\]

\[
\text{Specificity} = \frac{TN}{(FP + TN)}
\]

\[
\text{Dice} = \frac{2TP}{(2TP + FP + FN)}
\]

\[
\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)}
\]

\[
\text{Precision} = \frac{TP}{(TP + FP)}
\]

\[
\text{Recall} = \frac{TP}{(TP + FN)}
\]

It can be seen from Table 5 that the specific values of these 9 test images are all above 0.95, which is close to 1. This means that the missed diagnosis rate of the ground glass opaque area of COVID-19 is relatively low. Once the opaque area of ground glass appears in the CT image of COVID-19, the COVSeg-NET model can easily find it. At the same time, with the exception of image 2, image 6, image 8, and image 9, the Accuracy and Precision values of the predicted mask images of the other five images are basically above 0.95, which shows that the COVSeg-NET model is more accurate in capturing the foreground. Since the foreground area is the location of the lesion, it means that the model can detect the lesion area of the COVID-19 lung CT image well. Except for a few images, most of the predicted Dice values can still reach about 0.8, and it can be seen that the COVSeg-NET model has a good overall effect on the segmentation of COVID-19 lung CT images. In addition, due to the small number of COVID-19 lung CT images used for training, the amount of information contained is limited, and the sensitivity and recall rate of the COVSeg-NET model is low. Therefore, the model still needs to be improved, especially for the specific morphological segmentation task of the lesion area. In the later stages, if we can obtain more useful/high-quality clinical COVID-19 lung CT images, and at the same time make further improvements to the model structure, I believe we can achieve better results.

After evaluating the segmentation prediction results of a single COVID-19 CT image, we use the all test dataset to comprehensively evaluate the performance of covseg network. Here we use some experimental results of Mohamed et al. to compare with our experiments, as shown in Table 6.

Through comparison, it is found that when the quantity and quality of the data set are roughly the same, the COVSeg-NET model proposed in this study has better Dice coefficient and Specificity when segmenting the GGO area in the CT images of COVID-19, but the Sensitivity value is insufficient compared with other models in Table 6. A small sensitivity means that the model has a weak ability to capture the GGO area completely. Although the model can capture the specific location of the lesion, it cannot predict the specific shape of the lesion. The specificity is close to 1, which indicates that the COVSeg-NET model can almost capture the background well, which means that the model is almost not misdiagnosed. If there is no GGO region in the CT image of the patient’s lung, the model can make almost accurate judgment without mistaking the healthy region for GGO region. The Dice coefficient is actually the harmonic average of Precision and Recall, so it can reflect the overall performance of the model to a large extent. Therefore,
Nine test images use the COVSeg-NET segmentation model to obtain the prediction mask map and the real mask map to overlap, thereby obtaining TP, FP, TN, FN indicators.
on the whole, the COVSeg-NET model proposed in this study has better performance in segmenting the GGO region in CT images of COVID-19 than other models appearing in Table 6. The COVSeg-Net model is especially suitable for MIS targets with a small number of samples and a small problem space in medical images. Taking into account the limitations of the computational resources (mainly memory space) of this experiment, this study uses a fixed image size as the input of the model (that is, 128 x 128 pixels). But for images with very large problem space, this model is difficult to fully describe the relationship in the graph.

In recent years, nanotechnology has shown great potential application value in the medical field. By applying nano-imaging technology to magnetic resonance imaging and CT imaging, the quality of medical images can be greatly improved. Nanotechnology-based fluorescent particles can show excellent performance in vivo imaging, and this application provides better penetration, sensitivity, and resolution for in vivo imaging. Wen et al. linked NIR-II Ag2S quantum dots with amphiphilic peptides (APP) and tumor-targeted RGD peptides to form nano-chain-APP-Ag2S-RGD. Thus, nanomaterials can accumulate in tumor tissues and display the boundaries of metastatic tumors in the liver and spleen in vivo with high resolution. The combination of nanotechnology and biosensors can provide powerful pathogen detection capabilities, and this method has also made better progress in COVID-19 detection. Therefore, nanotechnology can improve the quality of medical images, thereby increasing the possibility of using deep models for disease detection. At the same time, nanotechnology can also be used as a material for the treatment of human diseases. In short, the use of nanotechnology to assist diagnosis and treatment is an important topic and direction for medical treatment in the future.

**5 | CONCLUSION**

This study designed a deep neural network segmentation model called COVSeg-NET, which can segment the ground glass shadow area in the lung CT image of COVID-19. The COVSeg-NET model extracts image feature information through multiple down-sampling and up-sampling operations, and performs multi-scale information fusion of different levels of information, so as to achieve the effect of comprehensive analysis of global and local details in the image. The image set used in this study is selected from 929 COVID-19 CT images and masks of the shadow area of the ground glass, and the data scale is expanded by data amplification, and then used for the training of the COVSeg-NET model. After training, the COVSeg-NET model is used to predict the test image set. In the end, the segmentation accuracy of the COVSeg-NET model reached 83.3%. Compared with some recently proposed models for COVID-19 lung imaging lesion segmentation, the COVSeg-NET model showed better performance. Currently, using a smaller training set of COVID-19 lung CT images, the COVSeg-NET model has been able to obtain good segmentation results. At the same time, we found through experiments that the model can also achieve good results in segmenting other medical images such as melanoma, white blood cells, and
cervical images. In the next work, we will collect more useful clinical data and optimize the structure of the model to improve the quality of medical image analysis.

The COVSeg-NET model has greater advantages in execution efficiency, robustness and generalization capabilities, and is especially suitable for large-scale MIS. Once the model training is completed, a large number of MIS and image annotation can be completed in a short time. Therefore, the COVSeg-NET model is expected to replace subjective and expensive manual segmentation.

DATA AVAILABILITY STATEMENT
The data that support the findings of this study are available from [Kaggle]. Restrictions apply to the availability of these data, which were used under license for this study.

ORCID
XiaoQing Zhang https://orcid.org/0000-0002-1072-6652

REFERENCES
1. Chandraa TB, Vermaka K, Singhbk BK, et al. Coronavirus disease (COVID-19) detection in chest X-ray images using majority voting based classifier ensemble. Expert Syst Appl. 2021;165: 113909.

2. Muhammad E, Chowdhury H, Rahman T, Khandakar A, et al. Can AI help in screening viral and COVID-19 pneumonia? IEEE Access. 2020;8:132665-132676. https://doi.org/10.1109/ACCESS.2020.3010287.

3. Chung M, Bernheima A, Mei X, et al. CT imaging features of 2019 novel corona-virus (2019-nCoV). Radiology. 2020;295(1): 202-207.

4. Huang C, Wang Y, Li X, et al. Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. Lancet. 2020;395(10223):497-506. [Published correction appears in Lancet 2020, 395(10223):496].

5. Hansxiang G, Ying X, Sheng N, et al. Clinical imaging features of novel coronavirus pneumonia (COVID-19). Radiol Pract. 2020; 35(2):125-130.

6. Yang X, He X, Zhao J, et al. COVID-CT-dataset: A CT scan dataset about COVID-19. arXiv:2003.13865; 2020.

7. Ouchicha C, Ammor O, Meknassi M. CV2DNet: a novel deep learning architecture for detection of coronavirus (COVID-19) from chest x-ray images. Chaos Solitons Fractals. 2020;140: 110245.

8. Wang L, Lin ZQ, Wong A. COVID-net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest x-ray images. arXiv, March, 2020.

9. Hassantabar S, Ahmadi M, Sharifi A. Diagnosis and detection of infected tissue of COVID-19 patients based on lung X-ray image using convolutional neural network approaches. Chaos Solitons Fractals. 2020;140:110170.

10. Hemdan EE, Shouman MA, Karar ME Covidx-net: a framework of deep learning classifiers to diagnose COVID-19 in X-ray images. arXiv Preprint arXiv: 2003. 11055. March 24, 2020.

11. Abdel-Basset M, Chang b V, Hawasha H, Chakrabortytc RK, Ryanc M. FSS-2019-nCoV: a deep learning architecture for semi-supervised few-shot segmentation of COVID-19 infection. Knowl-Based Syst. 2021;212:106647.

12. Zheng C, Deng X, Qiang F, et al. Deep learning-based detection for COVID-19 from chest CT using weak label. medRxiv preprint. https://doi.org/10.1101/2020.03.12.20027185.

13. Oulefki A, Agaian S, Trontigirakul T, et al. Automatic COVID-19 lung infected region segmentation and measurement using CT-scans images. Pattern Recognit. 2021;114:107747. https://doi.org/10.1016/j.patcog.2020.107747.

14. Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. In: Proceedings of the 13th International Conference on Medical Image Computing and Computer-Assisted Intervention. Munich: Springer, 2015;234-241.

15. Liu W, Rabinovich A, Berg AC. ParseNet: looking wider to see better. arXiv preprint, arXiv: 1506. 04579; 2015.

16. Abadi M, Agarwal A, Barham P, et al. TensorFlow: large-scale machine learning on heterogeneous systems. http://tensorflow.org/.

17. Jia Y, Shlamar E, Donahue J, et al. Caffe: convolutional architecture for fast feature embedding http://arxiv.org/abs/1408.5093.

18. Keras, http://keras.io/.

19. Yu D, Eversole A, Seltzer M, et al. An introduction to computational networks and the computational network toolkit. Microsoft Research

20. Torch7, https://github.com/torch/torch7.

21. Chen T, Li M, Li Y, et al. MXNet: a flexible and efficient machine learning library for heterogeneous distributed systems. 2015.

22. Leaf, https://github.com/autumnai/leaf.

23. Theano, http://deeplearning.net/software/theano.

24. DeepLearning4J, https://deeplearning4j.org/.

25. Lasagne, https://github.com/Lasagne/Lasagne.

26. Neon, https://github.com/NervanaSystems/neon.

27. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. Paper presented at: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2015:3431-3440.

28. Chen LC, Zhu Y, Papandreou G, Schroff F, Adam H. Encoder-decoder with atrous separable convolution for semantic image segmentation. Paper presented at: Proceedings of the European Conference on Computer Vision (ECCV); 2018:801-818.

29. Roy AG, Siddiqui S, Pösterl S, Navab N, Wachinger C. ‘Squeeze & excite’ guided few-shot segmentation of volumetric images. Med Image Anal. 2020;59:101587.

30. Zhang C, Zhanjun G, Liang Y. Application of nanotechnology in medical treatment. Knowl Mod Phys. 2018;030(005): 20-25.

31. Guo Z, Cui Z. Fluorescent nanotechnology for in vivo imaging. WIREs Nanomed Nanobiotechnol. 2021:e1705.

32. Wen Q, Zhang Y, Li C, et al. NIR-II fluorescent self-assembled peptide nanochain for ultrasensitive detection of peritoneal...
metastasis. *Angew Chem (Int Ed. in English)*. 2019;58(32):11001-11006.

33. Cui F, Zhou HS. Diagnostic methods and potential portable biosensors for coronavirus disease 2019. *Biosens Bioelectron*. 2020;165:112349.

34. Guo J, Huang X, Ma X. Clinical identification of diabetic ketosis/diabetic ketoacidosis acid by electrochemical dual channel test strip with medical smartphone. *Sens Actuators*. 2018;275:446-450.

35. Seo G, Lee G, Kim MJ, et al. Rapid detection of COVID-19 causative virus (SARS-CoV-2) in human nasopharyngeal swab specimens using field-effect transistor-based biosensor. *ACS Nano*. 2020;14(4):5135-5142.

---

**How to cite this article:** Zhang XQ, Wang GY, Zhao S-G. COVSeg-NET: A deep convolution neural network for COVID-19 lung CT image segmentation. *Int J Imaging Syst Technol*. 2021;31(3):1071–1086. [https://doi.org/10.1002/ima.22611](https://doi.org/10.1002/ima.22611)