EAI Endorsed Transactions
on e-Learning

Research Article **EAI.EU**

Gray level co-occurrence matrix and Schmitt neural network for Covid-19 diagnosis

Pengpeng Pi[1,*]

[1]College of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, Henan 454000, P R China

Abstract

INTRODUCTION: When COVID-19 spreads to most of the world, chest CT imaging is widely regarded as a convenient and feasible method for the diagnosis of suspected patients. In the traditional diagnosis method, doctors and experts judge these CT images and draw conclusions. However, with the surge in the number of suspected patients, relying solely on traditional manual diagnosis methods can no longer meet people's demand for efficiency.
OBJECTIVES: A number of previous studies have shown that it is possible to use machine learning methods to help people diagnose suspected SARS-CoV-2 patients. However, the accuracy of the existing scheme still needs to be improved.
METHODS: In order to make a more accurate diagnosis of suspected patients with SARS-CoV-2, we proposed a new model. We first preprocessed the CCT images of the tested objects to seek for higher accuracy, then extracted the texture features from the processed CCT images, and finally divided the CCT images of the tested objects into two categories: sick and normal using Schmidt neural network.
RESULTS: The accuracy of the proposed model is (76.33±1.18%), while the accuracy of the existing model RBFNN, ELM-BA, WEBBO and GLCM-SVM are (73.45±0.69%), (74.88±0.86%), (70.48±0.81%) and (64.42±0.88%), respectively. Compared with the existing RBFNN, ELM-BA, WEBBO and GLCM-SVM models, the accuracy of our proposed model is 1.45% higher than that of the best ELM-BA model. More importantly, the proposed model has better stability.
CONCLUSION: The model we proposed is feasible for the corresponding diagnosis of suspected patients. This is not only conducive to timely treatment of patients, but more importantly, effective isolation of confirmed patients as soon as possible can prevent the further spread of the epidemic.

Keywords: Machine Learning, Schmitt neural network, SARS-CoV-2.

Received on 18 June 2021, accepted on 19 July 2021, published on 11 August 2021

doi: 10.4108/eai.11-8-2021.170668

1. Introduction

1.1. Background

In December 2019, the first case of SARS-CoV-2 was detected globally, and just a few months later, SARS-CoV-2 has spread to most countries and regions in the world [1]. On 11 February 2020, the novel coronavirus2019 coronavirus was officially named COVID-19, and on 11 March of the same year, the World Health Organization called the current COVID-19 a global pandemic [2]. This has caused panic among people all over the world. Some countries and regions have taken drastic measures to prevent the spread of the epidemic to their own countries, such as restricting people's entry and exit to the border, canceling international flights and other measures. According to the latest data from the World Health Organization on the epidemic: as of 16:45 on May 23, Central European Summer Time, there were 166,346,635 newly confirmed cases of coronary artery pneumonia and 3,449,117 deaths worldwide. On the 23rd, there were 563,692 new confirmed cases of coronary artery

[*]Corresponding author. Email: pipengpeng@home.hpu.edu.cn

pneumonia worldwide, and 11,548 new deaths, which are still increasing. The epidemic has had a huge impact on the world politically and economically, and has fragmented countless families [3].

Although now through the continuous efforts of researchers, some countries have developed a new crown vaccine. In some countries, the epidemic has been further controlled through vaccination [4]. However, a considerable part of countries and regions do not have the economic and technological strength to conduct vaccine research and development and vaccination [5]. Therefore, isolation of infected persons is still the main means to prevent the spread of the epidemic at this stage. Timely diagnosis of suspected patients can break the chain of transmission and is essential to prevent further large-scale spread of the epidemic [6, 7]. From the clinical observation of a large number of SARS-CV-2 patients, it is known that most infected patients will develop fever, dry cough, fatigue and other typical characteristics after the incubation period. However, there are a small number of infected people who still do not have any clinical symptoms after the incubation period [8]. In some cases, the incubation period is 3-14 days, but in a few cases, the incubation period is as long as 24 days. Therefore, COVID-19 virus is highly infectious, highly pathogenic and has a long incubation period [9]. Therefore, it is not advisable to rely only on routine symptoms to determine whether a patient has the disease. The identification of suspected COVID-19 patients is very difficult compared to those infected with conventional virus. At present, the most commonly used method for testing suspected SARS-CoV-2 patients is nucleic acid testing. This method first collects nasopharyngeal swabs from the subject, and then uses RNA reverse transcription and polymerase chain reaction (RT-PCR). The principle draws a conclusion whether the subject is sick or not [10]. When the test subject's nucleic acid test result is positive, we will classify it as a high-risk group and immediately isolate the subject and treat it accordingly. Conversely, when the nucleic acid test result is negative, we temporarily classify it as a low-risk group. However, according to existing studies, the results of nucleic acid testing cannot be regarded as the only criterion for whether the subject is ill. Secondly, when we conduct nucleic acid testing, we need specialized testing personnel and reagents. In some economically underdeveloped countries and regions, they lack relevant medical personnel and corresponding reagents. Normally, the results of nucleic acid testing are relatively slow. During this period, the subject is at risk of spreading the epidemic. Therefore, we urgently need an accurate and efficient detection method for the tested object [11]. Since the lung images of SARS-CoV-2 patients have features such as pulmonary sclerosis and ground glass shadows, this is very helpful for us to use the CCT images of the tested subjects for corresponding diagnosis. Moreover, compared with nucleic acid detection methods, it is faster and safer to use the CCT image of the subject to perform the corresponding diagnosis. Therefore, chest imaging has become a better method to diagnose potential new coronary pneumonia [12].

1.2. Related work

Although compared with nucleic acid detection, it is safer and faster to use the CCT image of the subject for diagnosis, but it still has some shortcomings. When making a corresponding diagnosis, experts and doctors generally give corresponding conclusions. This method is more dependent on the doctor's current state. Some experts and scholars have begun to consider using artificial intelligence to help people diagnose CCT images of subjects. [13] The combination of wavelet entropy (WE) and biogeographic optimization (BBO) is used to diagnose SARS-CoV-2. [14] The gray level co-occurrence matrix (GLCM) and support vector machine (SVM) are used to classify the new SARS-CoV-2. [15] proposed a Radial Basis Function Neural Network (RBFNN) and applied it to the detection of human brain pathology. In this article, we use it as a comparative experiment in the detection of COVID-19. [16] combined extreme learning machine (ELM) with bat algorithm (BA). This ELM-BA is used as comparison basis in our experiment. [17] proposed an optimized method for the diagnosis of SARS-CoV-2, which is based on three-stage biogeography. [18] presented CCSHNet for COVID-19 diagnosis. [19] presented an attention network for COVID-19 (ANC) which can provide explainable diagnosis. [20] In order to find the influence of CNN's depth and fine-tuning degree on migration learning, a detailed comparison experiment was carried out using VGG-16 and VGG-19. The result shows that the deeper the depth, the better the result. [21] First, a data set containing 5000 images was collected, and four networks ResNet18, ResnNet50, SqueezeNet and DenseNet-121 were trained on this data set using transfer learning. Finally, a technique was used to generate a heat map of the lungs of people infected with COVID-19. [22] In this work, a new convolutional neural network, CapsNet, is proposed for corresponding detection of suspected SARS-CoV-2 patients. The two-class and multi-class accuracy of the model reached 97.24% and 84.22% respectively. [23] In this article, we introduce a convolutional neural network, COVID-Net, which specifically detects SARS-CoV-2 from the CXR images of the lungs of suspected patients. And generated a public data set containing 13,975 CXR images. [24] In this research, a new framework, CGNet, is proposed to classify CXR images into two categories: normal and diseased. The best accuracy of this model is 98.72%, the sensitivity is 100%, and the specificity is 97.95%. [25] This paper proposes a lightweight CNN structure based on the SqueezeNet network structure. This model is used to classify the CCT of suspected SARS-CoV-2 patients accordingly. The final accuracy of the model is 85.03%. [26] This research proposes a computer-aided diagnosis (CAD) system for SARS-CoV-2 detection. The system consists of feature extractors, classification methods and content-based image retrieval (CBIR). The accuracy of the model on the CT data set and CXR data set reached 93.20% and 99.38% respectively.

1.3. Our work

At present, there is an indisputable fact in the research field that most of our neural network training is based on gradient descent training. But algorithms based on gradient descent have the same shortcomings. All corresponding parameters need to be updated and adjusted during the training process, which will greatly reduce the training speed. Therefore, considering people's needs for efficiency and speed, we no longer use traditional neural networks based on gradient descent algorithms, but instead use Schmidt neural networks that can initialize weights and biases randomly. We first use the gray-level co-occurrence matrix to find the texture features of the preprocessed image, and then use the Schmidt neural network to classify it. After that, we compared our model with the existing RBFNN, ELM-BA, WEBBO, GLCM-SVM and other models. The results show that the performance of our proposed model has a significant improvement over the existing models. The main structure of this article is as follows: In the second section, we introduce the applied data set and the corresponding preprocessing work. In the third section, we describe the corresponding structure of the model and the main methods of application and evaluation indicators. In the fourth section, we summarize and state the experimental results accordingly. In the fifth section, we give the final conclusion and the defects of the model, as well as the work we need to do in the future.

2. Dataset

We obtained CCT scan images of a total of 284 study subjects from 142 patients with new coronary pneumonia and 142 healthy controls from local hospitals. The specific collection method is as follows: Philips Craft 64-slice spiral CT, KV: 120, Mas: 240, layer thickness 3mm, layer spacing 3mm, spacing 1.5: lung window (W: 1500HU, L: -500HU), mediastinal window (W: 350HU), L: 60HU). According to the lesion display, a lung window image with a layer thickness and a layer spacing of 1 mm was obtained. The patient is placed flat on the scanning instrument, and the patient is allowed to breathe deeply, and then scan from the top of the lung to the corner of the ribs.

We adopt a layer selection method for SARS-CoV-2 patients, choosing the layer with larger lesion area and more lesions. For healthy subjects, we can use images of any level, without the need to adopt a level selection method. Table 1 shows the demographic data of the subjects used in this study. We finally got a total of 680 CCT images including COVID-19 patients and healthy subjects. The resolution of the image is 1024×1024.

When we came up with two conclusions (Z_1, Z_2) for a CCT image, we asked senior doctors and experts to help us reach the final conclusion. Assuming that Y represents a CTT image scan, N means the conclusion given by each expert or doctor, and the final annotation conclusion M is determined by the following formula:

$$M(Y) = \begin{cases} N(Z_1) & N(Z_1)=N(Z_2) \\ NV(Nall) & otherwise \end{cases} \tag{1}$$

$$Nall = [N(Z_1), N(Z_1), N(Z_2)] \tag{2}$$

NV represents the majority vote, and $Nall$ represents the opinions given by three experts.

We set the original image data set as W_1. W_1 contains 320 CCT images of SARS-CoV-2 patients and 320 CCT images of healthy subjects.

$$W_1 = \{w_1(1), w_1(2), \ldots, w_i(i), \ldots w_i(640)\} \tag{3}$$

Fig 1 shows the original CT image of chest of a COVID-19 patient. We first processed the original image with grayscale, and only the brightness information in the image was retained. We then transformed the original data set W_1 with channel number 3 into the grayscale image set W_2 with channel number 1.

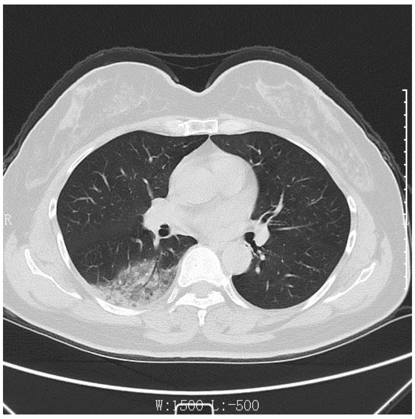


Fig 1. A raw COVID-19 image

$$W_2 = G(W_1|RGB \rightarrow Grayscale) = \{w_2(1), w_2(2), \ldots, w_2(i), \ldots w_2(640)\} \tag{4}$$

where G stands for gray scale operation.

Table 1. Details of the subjects tested in this study

	subjects	Images	Age
HC	142	320	24-88
COVID-19	142	320	22-75

In order to enhance the contrast of the gray image, we used the histogram stretching method to process the image accordingly. For each picture $w_2(i)$, $i = 1,2,\dots,640$, we use the following formula to calculate their minimum gray value $\sigma_{min}[w2(i)]$ and maximum gray value $\sigma_{max}[w2(i)]$.

$$\sigma_{min}[w2(i)] = min_{a,b=1}^{1024} w2(i|a,b) \quad (5)$$

$$\sigma_{max}[w2(i)] = max_{a,b=1}^{1024} w2(i|a,b) \quad (6)$$

Here a and b represent the pixel coordinates of the image $w_2(i)$, and the image $w_3(i)$ stretched by the histogram is obtained by the following formula.

$$w_3(i) = \frac{w_2(i)-\sigma_{min}[w2(i)]}{\sigma_{max}[w2(i)]-\sigma_{min}[w2(i)]} \quad (7)$$

Then the data set W_3 can be expressed as:

$$W_3 = HS(W_2) = \{w_3(1), w_3(2), \dots, w_3(i), \dots w_3(640)\} \quad (8)$$

After that, we trimmed the size of the image accordingly, and removed the edge and bottom blank areas of the image. Finally, we get the cropped data set W_4.

$$W_4 = C(W_3, [top, bottom, left, right]) = \{w_4(1), w_4(2), \dots, w_4(i), \dots w_4(640)\} \quad (9)$$

Here C represents the tailoring operation. We set the values of the four variables top, $bottom$, $left$ and $right$ to 150, that is, the part representing the image to be removed from the top, $bottom$, $left$ and $right$ is 150. Each image has now been cropped from 1024×1024 to 724×724.

Finally, we take the subsampling operation for all the images, and we get the resized image data set W_5.

$$W_5 = \downarrow (W_4|[256{,}256]) = \{w_5(1), w_5(2), \dots, w_5(i), \dots w_5(640)\} \quad (10)$$

Here ↓ represents the down-sampling operation. After the fourth step, we get the image data set W_5 with a number of 640, a channel number of 1, and a size of 256×256. Fig 2 shows the flow chart of our image preprocessing process in detail.

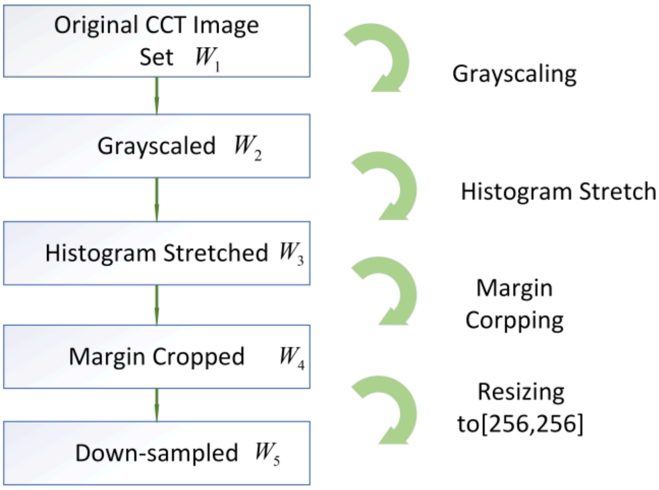


Fig 2. Image preprocessing process

Fig 3 shows two pre-processed examples. Table 2 shows the image size and required memory space after each pre-processing stage. We found that after these four pre-processing steps, each image required only 2.08% of the original memory space. Therefore, the experimental application of pre-processed images will save us a lot of memory resources. The efficiency of our experiments will also be greatly improved. We compared the memory space required by the final image with the original image. The comparison results are as follows:

$$CR_{storage}(i) = \frac{byte(w_5(i))}{byte(w_1(i))} = \frac{262{,}144}{12{,}582{,}912} = 2.083\% \quad (11)$$

$$CR_{size}(i) = \frac{size(w_5(i))}{size(w_1(i))} = \frac{65{,}536}{3{,}145{,}728} = 2.083\% \quad (12)$$

Table 2. Image size and required memory space

Preprocess	Symbol	Size	Storage
Original	$w_1(i)$	1024×1024×3=3,145,728	125,582,912
Grayscale	$w_2(i)$	1024×1024×1=1,048,576	4,194,304
HS	$w_3(i)$	1024×1024×1=1048,576	4,194,304
Crop	$w_4(i)$	724×724×1=524,176	2,096,704
DS	$w_5(i)$	256×256×1=65,536	262,144

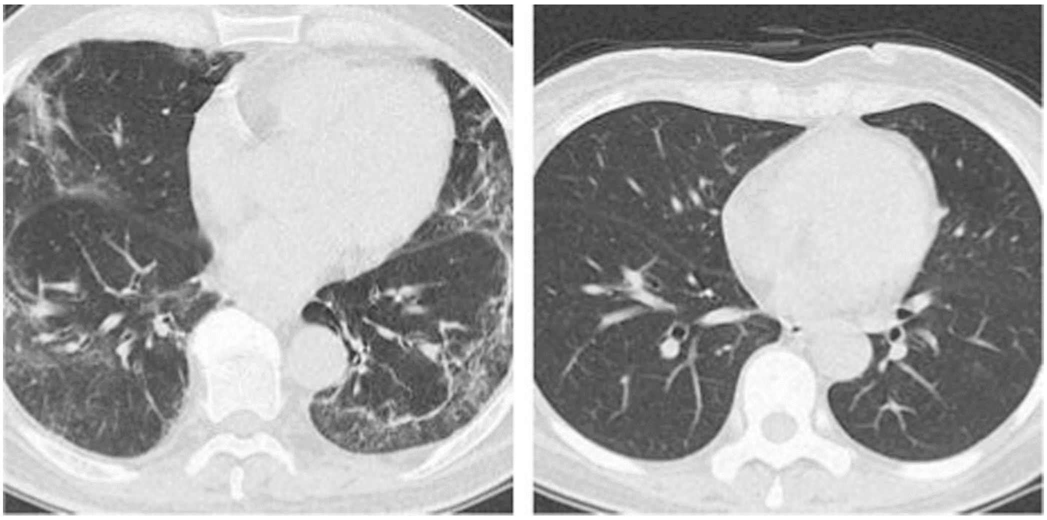


Fig 3. Preprocessed image sample

3. Methodology

In this section, we will elaborate on the proposed model. The principle of gray level co-occurrence matrix for image feature extraction and the basic concept of Schmidt neural network are included. In addition, this section also introduces the required ten-fold cross-validation methods and the indicators required for model performance evaluation.

3.1. GLCM

Feature extraction refers to a method of transforming a piece of data to find its typical features. In the classification research, in order to reduce the burden of the classifier, we will perform corresponding feature extraction operations. Extracting the texture feature information in the image is an excellent method for X-ray image feature extraction. Research on existing data shows that texture features are particularly suitable for X-ray image classification. The texture feature will also consider the quantization and positioning relationship of the pixel group with the same intensity. Any grayscale surface of an image can be viewed as a curved surface in three dimensions. In a three-dimensional space, two pixels are separated by a distance. They may have the same grayscale or different grayscales. GLCM [27] is to start from the pixel of image (x, y) whose gray level is i, and to count the probability $P(i, j, \delta, \theta)$ of the simultaneous occurrence of pixel $(x + d, y + l)$ whose distance is δ and gray level is j. This is shown in Fig 4. Using mathematical formula, it can be written as:

$$P(i,j,\delta,\theta) = \{[(x,y),(x+d,y+l)] \mid (f(x,y) = i, f(x,y) = i, f(x+d,y+l) = j; \\ x = 0,1,\dots,N\_x - 1; y = 0,1,\dots,N\_y - 1)\} \quad (13)$$

Where: $i, j = 0,1,\dots,L-1$; x, y are the pixel coordinates in the image; L is the number of pixel gray levels; N_x represents the number of image columns, and N_y represents the number of rows of the image.

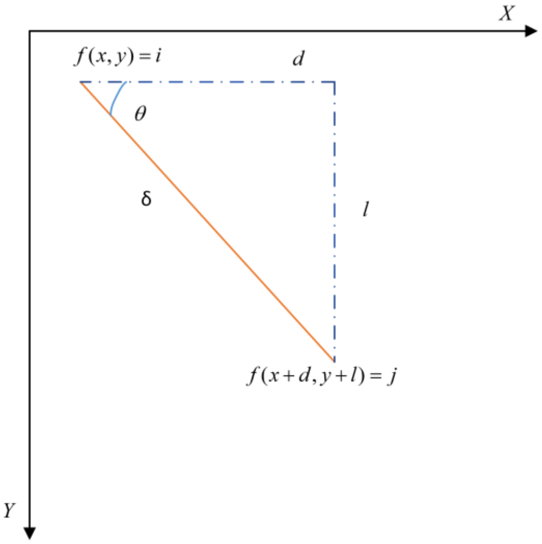
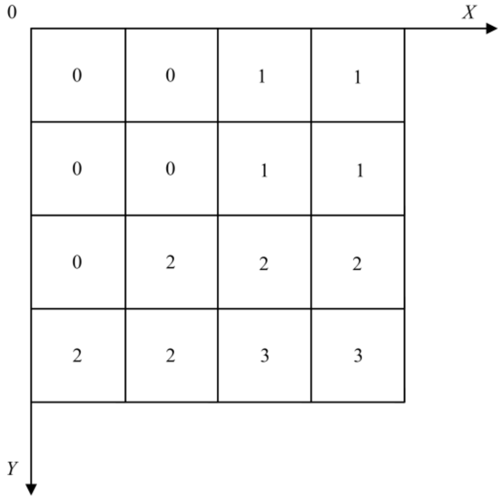


Fig 4. Sample graph of Gray Level Co-occurrence Matrix

It can be seen from the above definition that the elements in the ith row and jth column of the GLCM constitute the frequency of occurrence of a pixel pair. The gray value of one pixel of this pixel pair is j, and the value of the other pixel is i. The angle between them is θ, and the interval is δ. Here θ is the included Angle between the X-axis and the two pixels in the clockwise direction [28], and θ is generally 0°, 45°, 90° and 135°. Obviously, if $d = 1$ and $l = 0$, then θ is 0°, $d = 1$, $l = 1$, $\theta = 45^\circ$; $d = 0$, $l = 1$, $\theta = 90^\circ$; if $d = -1$

and $l = 1$, then $\theta = 135^{\circ}$. The value of δ is related to the image, which is generally determined by experiment.

For example, for the image shown in Fig 5, take the interval $d = 1$, and the gray co-occurrence matrix in all directions is shown in Fig 6.



0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

Fig 5. Texture image

$$P_{0^{\circ}} = \begin{bmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix} \qquad P_{45^{\circ}} = \begin{bmatrix} 2 & 1 & 3 & 0 \\ 1 & 2 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{bmatrix}$$

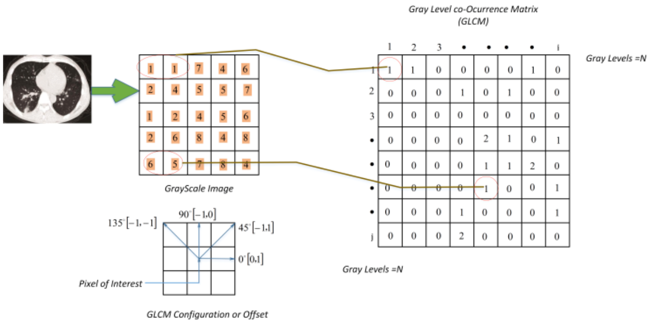
$$(a) \qquad\qquad (b)$$

$$P_{90^{\circ}} = \begin{bmatrix} 6 & 0 & 2 & 0 \\ 0 & 4 & 2 & 0 \\ 2 & 2 & 2 & 2 \\ 0 & 0 & 2 & 0 \end{bmatrix} \qquad P_{135^{\circ}} = \begin{bmatrix} 4 & 1 & 0 & 0 \\ 1 & 2 & 2 & 0 \\ 0 & 2 & 4 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$(c) \qquad\qquad (d)$$

Fig 6. Gray-level co-occurrence matrix in all directions

We created the corresponding GLCM based on the CCT image, the offset in this process is determined and the creation process is shown in Fig 7. We first convert the preprocessed image to its corresponding grayscale image [29], that is, the intensity of each pixel with a depth of N bits ranging from 0 to 255. We then define the offset of the symbiotic comparison. Since the gray level co-occurrence matrix is two-dimensional and our image has only 8 gray levels at maximum, this matrix is 8×8 matrix. The gray co-occurrence matrix reflects the gray direction of the image. It also in turn reflects the comprehensive information of gray changes in adjacent intervals. We can use it as information to analyze the image base and arrangement structure. However, we generally do not directly calculate the GLCM as the characteristic quantity of texture analysis. When we extract texture features, we generally use the GLCM, which is called quadratic statistics. (14)-(18) Give the calculation equation of each characteristic quantity. Table 3 provides a detailed explanation of the symbols in the example.



1	1	7	4	6
2	4	5	5	7
1	2	4	5	6
2	6	8	4	8
6	5	7	8	4

	1	2	3	•	•	•	•	i
1	1	1	0	0	0	0	1	0
2	0	0	0	1	0	1	0	0
3	0	0	0	0	0	0	0	0
•	0	0	0	0	2	1	0	1
•	0	0	0	0	1	1	2	0
•	0	0	0	0	1	0	0	1
•	0	0	0	1	0	0	0	1
j	0	0	0	2	0	0	0	0

Fig 7. Example of GLCM Creation

Table 3. Example symbol interpretation

Where:

N = Represents the total number of gray levels in the image.
i = Represents the ith position of the $N_x N$ GLCM.
j = Represents the jth position of the $N_x N$ GLCM.
$p(i, j)$ = gives the gray level co-occurrence value of the $p(i, j)th$ position in the GLCM.

CON: A measure of the result of comparing a pixel value with the pixel value of the surrounding area. This metric value also reflects the clarity of the image and the depth of the image texture. When the sharpness of the image is higher, the texture of the image will be deeper. When the sharpness of the image is low, the texture of the image will be lighter.

$$Contrast = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i - j)^2 p(i, j) \qquad (14)$$

COR: Used to express the degree of local correlation in grayscale images. When the value distribution of the gray-level co-occurrence matrix in a certain image range is relatively uniform, its correlation will be greater. Conversely, when the value distribution of the gray-level co-occurrence matrix in a certain image range is chaotic, its correlation will be smaller. When texture features appear in a certain direction, the correlation in this direction is greater than the correlation in other directions.

$$Correlation = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{(i-\mu_i)(j-\mu_j)p(i,j)}{\sigma_i \sigma_j} \qquad (15)$$

ASM: It is a measure of whether the grayscale distribution in the image is uniform, and it is also a measure of the depth of texture in the image. Its calculation method is to first square all the elements in the gray-level co-occurrence matrix, and then add them up to get the energy of the image. If the gray value fluctuation range in the matrix is relatively small, then the energy value of the image is relatively large, on the contrary, if the gray value fluctuation range in the matrix is relatively large, then the energy value of the image is relatively small.

$$ASM = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left(p(i,j)\right)^2 \quad (16)$$

Entropy: Entropy is a measure of the non-uniformity of image texture. When the element values of the matrix are randomly distributed, the entropy value of the image is larger. Conversely, when the element values of the matrix are distributed regularly, the entropy value of the image is smaller.

$$Entropy = -\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} p(i,j)\log\left(p(i,j)\right)(17)$$

IDM: Reflects the clarity and regularity of the texture. When the texture of the image is clearer, the regularity is stronger, and the texture is better than the description, the greater the value of IDM. On the contrary, when the texture of the image is messy and the texture is more difficult to describe, the value of IDM is smaller.

$$Inverse\ Different\ Moment\ or\ Homogeneity = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\frac{1}{1+(i-j)^2}p(i,j) \quad (18)$$

When we calculate the texture features of these images, the means μ_i μ_j and standard deviations σ_i σ_j of i and j have to be calculated as follows:

$$\mu_i = \sum_{i=1}^{N} ipx(i) \quad (19)$$

$$\mu_j = \sum_{i=1}^{N} ipy(i) \quad (20)$$

$$\sigma_i = \sqrt{\sum_{i=1}^{N} p_x(i)(i-\mu_i)^2} \quad (21)$$

$$\sigma_j = \sqrt{\sum_{i=1}^{N} p_y(i)\left(i-\mu_j\right)^2} \quad (22)$$

$p_x(i)$ is the marginal probability of i in p_x and $p_y(j)$ is the marginal probability of j in p_y.

$$p_x(i) = \sum_{j=1}^{N} p(i,j) \quad (23)$$

$p_x(i)$ is the ith entry in the marginal probability matrix obtained by adding the rows of $p(i,j)$.

$$p_y(j) = \sum_{i=1}^{N} p(i,j) \quad (24)$$

$p_y(j)$ is the jth entry in the marginal probability matrix obtained by summing the columns of $p(i,j)$.

3.2. Schmidt Neural Network

Similar to the traditional single hidden layer neural network structure, the extreme learning machine has only a three-layer structure, including an input layer, a hidden layer in the middle, and an output layer at the end. However, compared with the traditional neural network based on gradient descent algorithm, the extreme learning machine has a very different parameter update. The basic core of the traditional backward propagation algorithm is to use the gradient descent mechanism to update all the corresponding parameters each time during the training process [30]. This training method is very time-consuming and cannot meet people's high-efficiency needs. The extreme learning machine uses a new training algorithm, which randomly generates input weights and deviations during the training process, and at the same time derives the output weights from the principle of the generalized inverse matrix. Therefore, compared with the traditional neural play network based on gradient descent algorithm, it greatly shortens the time required for training while ensuring that the accuracy is almost unchanged. The structure of ELM is shown in Fig 8. The structure of Schmidt Neural Network (SNN) is similar to that of Extreme Learning Machine (ELM), and the principle of training is also the same. The only difference between the two is that the output deviation of the extreme learning machine is always 0. But the 1 output deviation of the Schmidt neural network may not be 0. The structure of the Schmidt neural network is shown in Fig 9.

We assume that the number of training sets is N, and each sample in the training set can be expressed as (X_i, t_i), where $X_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in R^n$, $t_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T \in R^m$, X_i represents the ith data, and t_i represents the label corresponding to the ith data. For a neural network, we can simply regard it as a function. There is a hidden layer from the input layer to the corresponding output layer, the hidden layer is a fully connected layer, and there are L nodes in the hidden layer. When we are given N data set samples, the corresponding output of the hidden layer is:

$$\sum_{i=1}^{L}\beta_i g\left(W_i \cdot X_j + b_i\right) = o_j,\ \ j = 1, \dots, N \quad (25)$$

The hidden layer contains several non-linear activation functions that can be different. The input of the input layer is multiplied by its weight, and then the corresponding deviation is added to transform the non-linear activation function to obtain the output of the hidden layer. Where in, $W_i = [w_{i,1}, w_{i,2}, \dots, w_{i,n}]^T$ represents the set of weights corresponding to the input layer, β_I represents the set of corresponding hidden layer output weights, and b_i represents the corresponding set of deviations corresponding to the input layer. $W_i \cdot X_j$ is the inner product of W_i and X_j. The $g(x)$ corresponds to the activation function of the hidden layer. The activation function is generally nonlinear. Common activation functions include Sigmoid function, Gaussian function, etc.

In order to obtain β_I that has a good effect on the training sample set, we must ensure that the error reaches its minimum during the training process. We can set the square difference between the label value t_j and the output o_j of the output layer as the objective function. To minimize the error in the training process, we only need to minimize the value of the objective function. According to the definition, the objective function can be written as:

$$\sum_{j=1}^{N} ||o_j - t_j|| = 0 \quad (26)$$

So there are β_i, W_i and b_i, such that

$$\sum_{i=1}^{L}\beta_i g\left(W_i \cdot X_j + b_i\right) = t_j,\ \ j = 1, \dots, N \quad (27)$$

It can be represented matrices as:

$$H\beta = T \tag{28}$$

H represents the output matrix corresponding to the hidden layer, T represents the target matrix of the corresponding training data, and β represents the weight of the output corresponding to the hidden layer.

$$H(W_1, \dots, W_L, b_1, \dots, b_L, X_1, \dots, X_L) = \begin{bmatrix} g(W_1 \cdot X_1 + b_1) & \cdots & g(W_L \cdot X_1 + b_L) \\ g(W_1 \cdot X_N + b_1) & \cdots & g(W_l \cdot X_N + b_L) \end{bmatrix}_{N\times L} \tag{29}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ . \\ . \\ . \\ \beta_L^T \end{bmatrix}_{L\times m} \quad T = \begin{bmatrix} T_1^T \\ . \\ . \\ . \\ T_N^T \end{bmatrix}_{N\times m} \tag{30}$$

In order to train the entire single hidden layer neural network accordingly, we hope to obtain$\widehat{W}_\iota$, $\widehat{b}_\iota$ and$\widehat{\beta}_\iota$, such that:

$$\left\|H\left(\widehat{W}_\iota, \widehat{b}_\iota\right)\widehat{\beta}_\iota - T\right\| = min_{W,b,\beta}\|H(W_i, b_i)\beta_i - T\| \tag{31}$$

In this formula, $i = 1, \dots, L$, which means that we bring the value of the loss function to its corresponding minimum value.

$$E = \sum_{j=1}^{N}\left(\sum_{i=1}^{L}\beta_i g\left(W_i \cdot X_j + b_i\right) - t_j\right)^2 \tag{32}$$

We transform the equation into: $H\beta = T$, β can be written as:

$$\hat{\beta} = \bar{H}T \tag{33}$$

$\bar{H}$ is the Moore-Penrose inverse of the matrix. The output of SNN can be obtained by:

$$\sum_{i=1}^{L}\left[\beta_i g\left(W_i \cdot X_j + b_i\right)\right] + b = o_j, \quad j = 1, \dots, N \tag{34}$$

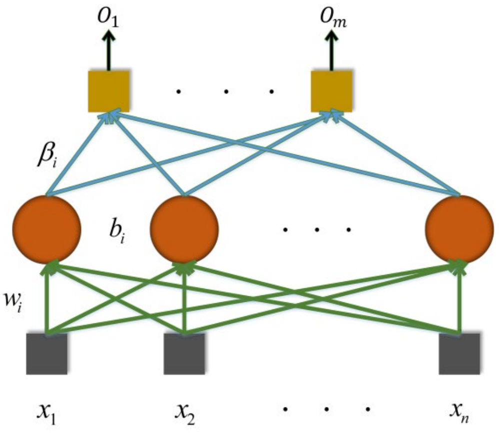


Fig 8. Structure of ELM

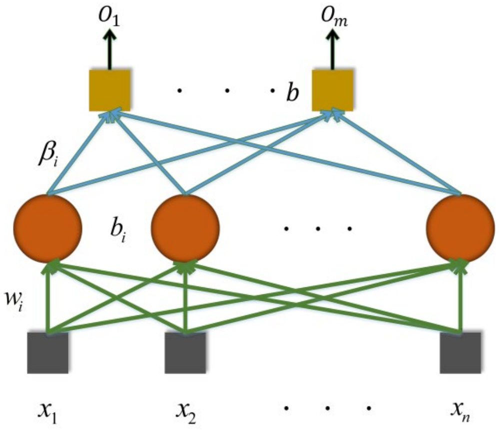


Fig 9. Structure of SNN

3.3. K-fold Cross Validation

When we train the model accordingly, the data set will be divided into two parts: training set and test set. The training set is responsible for training and updating the parameters of the model, and the test set is responsible for the final model performance test part [31]. However, due to the particularity of the research field, the medical field often faces the situation of missing data sets or insufficient data sets. If you still divide the data set into two major sections according to the original method, you will face the problem of model overfitting [32]. That is, the model performs well in the training set, but the performance in the test set is really unsatisfactory. But if the data in the test set is divided into the training set, this will improve the accuracy of the model. But the final accuracy of the model cannot be guaranteed. When we give the model a new test set, the model's fitting ability may be poor.

In order to solve this problem, we generally separate a part of the training set data as a validation set. The role of the validation set is to make corresponding choices for the generated models. At the same time, in order to prevent the problem of over-fitting, we will use the k-fold cross-validation method to solve it. The principle of this method is to evenly divide the data set into k equal parts, and take one of them as the verification set each time, and then the remaining k-1 parts as the corresponding training set data. Finally, repeat k times. In this case, each part of the data will eventually serve as the data part of a part of the validation set. We will eventually retain the average of the k results as the final result. In this way, we can effectively result in the problem of model overfitting.

One of the methods we often use to test the accuracy of algorithms is the ten-fold cross-validation method. Similar to the principle described above, we will first divide the data set into ten equal parts evenly, and each time nine of them will be used as training set to participate in the updating of model parameters, and the remaining one will be used as verification set to participate in the verification of the model. We will repeat this step ten times, so that each part of the dataset has a chance to act as a validation set. Finally, we keep the average of these ten results. There is some theoretical evidence that dividing the data set into 10 segments yields the best error estimates.

In this experiment, we used a ten-fold cross-validation method to evaluate the model. Fig 10 is a detailed illustration of the ten-fold cross-validation method.

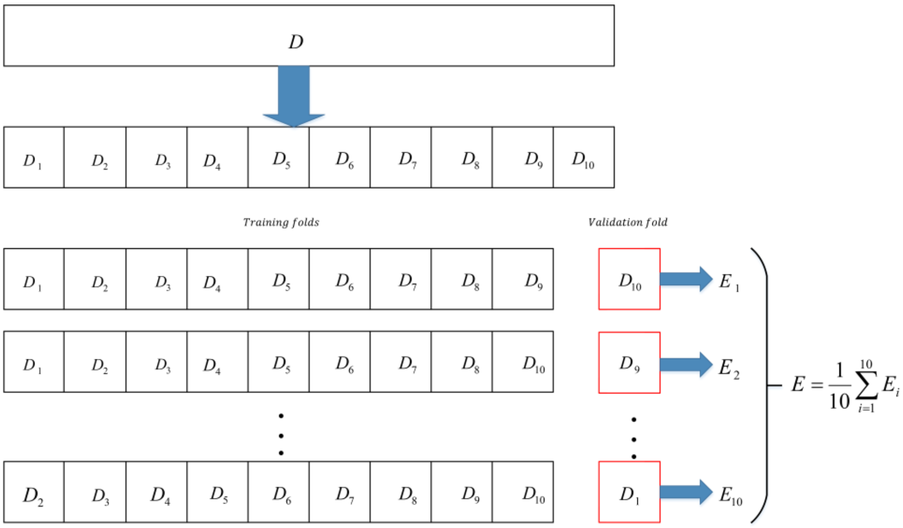

D
D1 D2 D3 D4 D5 D6 D7 D8 D9 D10
Training folds
Validation fold
E1
E2
E10
$E = \frac{1}{10}\sum_{i=1}^{10} E_i$

Fig 10. Schematic diagram of the ten-fold cross-validation method

3.4. Measurement

In order to verify whether our proposed model has the meaning of existence, we use the sensitivity, specificity, precision, accuracy, MCC, F1-Score and FMI as the indicators of our model performance measurement. In order to obtain these indicators, we need to use a visualization tool: confusion matrix. Its basic principle is to compare the predicted result with the actual result and produce four corresponding results. A true positive means that the predicted result is positive, and the actual result is also positive. Conversely, a false positive means that the predicted result is positive, but the actual label of the sample is negative. True negative means that the predicted result is negative, and the label of the actual sample is also negative. On the contrary, false negative means that the result predicted by the model is negative, but the label of the actual sample is positive. When we accumulate all false negative and false positive results, it represents all the parts of the sample that were predicted incorrectly. When we accumulate all true positive and true negative results, it represents all the parts of the sample that are predicted to be correct [33]. In the confusion matrix, the column of the matrix represents the collection of all the attribute samples predicted to be the column, and the row of the matrix represents the label that the row of samples should correspond to.

$$\begin{bmatrix} True\quad Positive(TP) & False\quad Negative(FN) \\ False\quad Positive(FP) & True\quad Negative(TN) \end{bmatrix} \quad (35)$$

More advanced classification indicators can be obtained from the confusion matrix: Accuracy, Precision, Specificity, Sensitivity, MCC, F1-score and FMI. For the dichotomy problem, the sample can be divided into: $total\ number\ of\ samples = TP + FP + TN + FN$ according to the combination of its real category and model predicted category.

1. Accuracy: It is often used to indicate the basic indicator that the model can make accurate predictions. We first add up the total number of true positive samples that the model can predict correctly plus the total number of true negative samples that the model can correctly predict, and then compare the total number of samples. The value obtained is the ability value of the sample to correctly predict the sample.

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \times 100\% \qquad (36)$$

2. Sensitivity: We need the model to be able to maximize the judgment of all positive samples as positive [34]. Therefore, the sensitivity is the ratio of the total number of samples that the model judges to be true positives to the total number of positive samples.

$$Sensitivity = \frac{TP}{TP+FN} \times 100\% \qquad (37)$$

3. Specificity: Contrary to sensitivity, specificity represents the model's ability to maximize the ability to identify true negatives in negative samples. Therefore, the specificity calculation formula is the total number of samples identified as true negative divided by the total number of samples actually labeled as negative.

$$Specificity = \frac{TN}{FP+TN} \times 100\% \qquad (38)$$

4. Precision: The precision of the model represents the ability of the model to truly identify true positive samples.

The calculation formula of precision is the ratio of the total number of true positive samples to the total number of positive samples judged by the model. The larger the value of precision, the stronger the ability to truly identify true positive samples.

$$Precision = \frac{TP}{TP+FP} \times 100\% \qquad (39)$$

5. MCC: In the application of machine learning, MCC is generally used to measure the classification performance in the case of dichotomy. This holds true even when the sample sizes of the two categories are significantly different.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)\times(TP+FN)\times(TN+FP)\times(TN+FN)}} \times 100\% \qquad (40)$$

6. F1-score: It is a common indicator to measure the performance of a classification model. Its maximum value is 1, and its minimum value is 0.

$$F1 = \frac{2TP}{2TP+FP+FN} \times 100\% \qquad (41)$$

7. The Fowlkes-Mallows score FMI [35] is defined as the geometric mean of the pairwise precision and recall.

$$\mathrm{FMI} = \frac{TP}{\sqrt{(TP+FP)(TP+FN)}} \times 100\% \qquad (42)$$

4. Experiment Result and Discussions

4.1. Statistical Analysis

In this experiment, we cooperated with local institutions and recruited a total of 284 volunteers to participate in this research. Our final data set obtained 640 lung CT images of the subject, including 320 lung CT images of SARS-CoV-2 infected persons and 320 lung CT images of healthy subjects. After that, in order to be more helpful to the experiment, we carried out preprocessing operations on these images. In the experiment process, the texture feature of the processed image is extracted through the generated GLCM. Finally, the extracted texture feature is classified by Schmidt neural network to obtain the experimental result. In order to evaluate the performance of our proposed model, we obtain the sensitivity, specificity, precision, accuracy, F1-score, MCC and FMI indicators of the model through the visual confusion matrix. The detailed data is shown in Table 4. The data shows: the average values of sensitivity, specificity, precision, accuracy, f1-score, MCC, and FMI are (74.66±1.87%), (78.00±1.29%), (77.24±1.15%), (76.33) ±1.18%), (75.92±1.31%), (52.70±2.34%), (75.93±1.30%). For easy observation, Fig 11 shows a line graph of the changes in the model's overall performance.

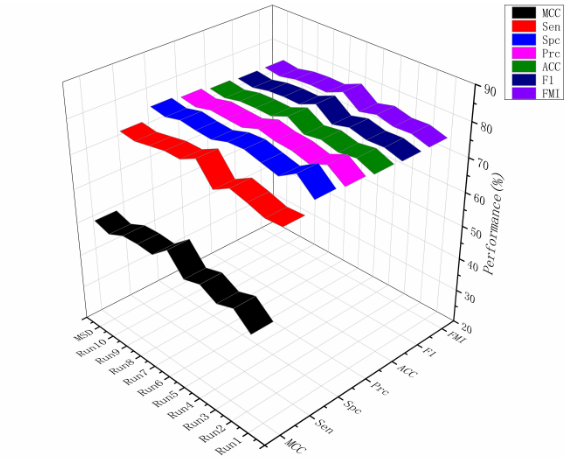


Fig 11 Model performance change trend

Table 4. Results of GLCM-SNN

Run	Sen	Spc	Prc	Acc	F1	MCC	FMI
1	72.81	75.62	74.92	74.22	73.85	48.46	73.86
2	72.50	80.00	78.38	76.25	75.32	52.65	75.38
3	74.06	76.88	76.21	75.47	75.12	50.96	75.13
4	75.31	78.12	77.49	76.72	76.39	53.46	76.39
5	72.50	79.38	77.85	75.94	75.08	52.00	75.13
6	78.44	79.06	78.93	78.75	78.68	57.50	78.68
7	75.94	77.50	77.14	76.72	76.54	53.44	76.54
8	75.94	77.81	77.39	76.88	76.66	53.76	76.66
9	75.00	78.44	77.67	76.72	76.31	53.47	76.32
10	74.06	77.19	76.45	75.62	75.24	51.28	75.25
MSD	74.66 ±1.87	78.00 ±1.29	77.24 ±1.15	76.33 ±1.18	75.92 ±1.31	52.70 ±2.34	75.93 ±1.30

4.2. SNN against Fully-connected layer

We replaced the SNN of this model with the full-connection layer network, and the results after ten times of ten-fold cross-validation are shown in Table 5. The results showed: the average sensitivity, specificity, precision, accuracy, F1-score, MCC and FMI were (74.03 ± 2.70%), (77.69 ± 1.45%), (76.86 ± 0.87%), (75.86 ± 1.00%), (75.39 ± 1.40%), (51.79 ± 1.94%), and (75.42 ± 1.36%). Before replacement, the model specificity, specificity, precision, accuracy, F1-score, MCC, and FMI values were (74.66 ± 1.87%), (78.00 ± 1.29%), (77.24 ± 1.15%), (76.33 ± 1.18%), (75.92 ± 1.31%), (52.70 ± 2.34%), and (75.93 ± 1.30%), respectively. It can be found that the proposed model is superior to the replaced model in terms of sensitivity, specificity, precision, accuracy, F1-score, MCC and FMI. Therefore, compared with the traditional neural network, SNN not only has a faster learning speed. In some specific tasks, the comprehensive performance of SNN is also better than that of the traditional neural networks. Fig 12 shows the trend chart of each index in detail.

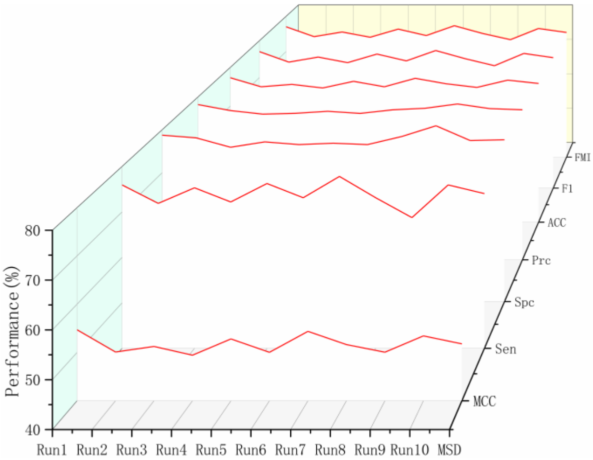


Fig 12. A comprehensive performance trend for SNN with one fully-connected layer

Table 5. Results of GLCM-FCL

Run	Sen	Spc	Prc	Acc	F1	MCC	FMI
1	75.94	78.75	78.14	77.34	77.02	54.71	77.03
2	71.88	78.12	76.67	75.00	74.19	50.10	74.23
3	75.31	75.94	75.79	75.62	75.55	51.25	75.55
4	72.19	77.19	75.99	74.69	74.04	49.44	74.06
5	76.25	76.56	76.49	76.41	76.37	52.81	76.37
6	73.12	76.88	75.97	75.00	74.52	50.04	74.54
7	77.81	76.56	76.85	77.19	77.33	54.38	77.33
8	73.12	78.44	77.23	75.78	75.12	51.64	75.15
9	68.75	80.94	78.29	74.84	73.21	50.06	73.37
10	75.94	77.50	77.14	76.72	76.54	53.44	76.54
MSD	74.03 ±2.70	77.69 ±1.45	76.86 ±0.87	75.86 ±1.00	75.39 ±1.40	51.79 ±1.94	75.42 ±1.36

4.3. Comparison with State-of-the-art approaches

We use RBFNN model, ELM-BA model, WEBBO model and GLCM-SVM model to carry out comparative experiments. The detailed data of the experiment are shown in Table 6. Fig 13 is a three-dimensional display of the performance of several models. It can be found that the model we proposed performs best in most of the indicators. In terms of sensitivity, it was 1.72% higher than WEBBO, and in terms of specificity, it was 0.62% higher than GLCM-SVM. In terms of precision and accuracy, compared with GLCM-SVM, the results are improved by 1.02% and 1.45%, respectively. In terms of F1-score and FMI, our model also increased by 2.47% and 1.68% respectively compared to GLCM-SVM. Compared with the above model, our proposed model uses the gray level co-occurrence matrix for image feature extraction. Compared with the above feature extraction method, this method can more comprehensively reflect the image direction, adjacent interval, change range and other comprehensive information. Image information. After that, we used Schmidt neural network to classify the corresponding features. Due to the random initialization of weights and biases, the network not only has a good improvement in classification effect compared with traditional forward neural networks, but also has a considerable improvement in classification efficiency.

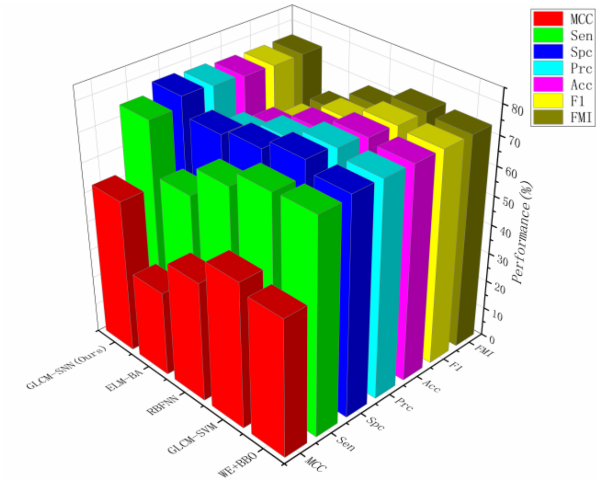


Fig 13. Comparison with State-of-the-art approaches

Table 6. 10 runs of 10-fold cross validation

Method	Sen	Spc	Prc	Acc	F1	MCC	FMI
WE+BBO [13]	72.94 ±0.96	73.97 ±1.02	73.70 ±0.79	73.45 ±0.69	73.31 ±0.71	46.91 ±1.38	73.32 ±0.71
GLCM-SVM [14]	72.38 ±2.68	77.38 ±1.96	76.22 ±1.21	74.88 ±0.86	74.21 ±1.25	49.85 ±1.70	74.25 ±1.21
RBFNN [15]	66.97 ±2.42	74.00 ±2.14	72.07 ±1.25	70.48 ±0.81	69.39 ±1.18	41.10 ±1.60	69.46 ±1.12
ELM-BA[16]	56.91 ±1.21	71.94 ±2.17	67.01 ±1.52	64.42 ±0.88	61.53 ±0.77	29.19 ±1.88	61.74 ±0.77
GLCM-SNN (Ours)	**74.66** **±1.87**	**78.00** **±1.29**	**77.24** **±1.15**	**76.33** **±1.18**	**75.92** **±1.31**	**52.70** **±2.34**	**75.93** **±1.30**

5. Conclusion

In this study, we introduced a model for COVID-19 detection. We first use the GLCM to extract the features of the preprocessed CCT image, and then use SNN to classify the image with the extracted features. We divide the images into two categories, one is the CCT of HC, and the other is the CCT of people infected with COVID-19. After that, in order to verify the effectiveness of our proposed model, we conducted comparative experiments with several existing methods: RBFNN, ELM-BA, WEBBO and GLCM-SVM. The experimental results show that the proposed model is not only the best in terms of specificity, accuracy, and precision than the existing models. In addition, in terms of model stability, our proposed model also performs best.

In the field of medical imaging, due to special reasons, the lack of data sets is a common phenomenon. Without sufficient data sets for corresponding training, the model cannot achieve the desired effect. In the follow-up work, we will collect more CCT images of COVID-19 patients to expand the data set. In addition, we will continue to improve the model, or develop a model with higher accuracy for the diagnosis of COVID-19, to help all mankind overcome this difficulty as soon as possible.

References

[1] P. J. Hotez, A. Fenwick, and D. Molyneux, "The new COVID-19 poor and the neglected tropical diseases resurgence," *Infectious Diseases of Poverty,* vol. 10, p. 3, Article ID: 10, 2021.

[2] S. H. Y. Yuan, "Introduction to the Special Issue on Computer Modelling of Transmission, Spread, Control and Diagnosis of COVID-19," *Computer Modeling in Engineering & Sciences,* vol. 127, pp. 385-387, 2021.

[3] A. Bracci, M. Nadini, M. Aliapoulios, D. McCoy, I. Gray, A. Teytelboym*, et al.*, "Dark Web Marketplaces and COVID-19: before the vaccine," *Epj Data Science,* vol. 10, p. 26, Article ID: 6, 2021.

[4] J. Samal, "Impact of COVID-19 infodemic on psychological wellbeing and vaccine hesitancy," *Egyptian Journal of Bronchology,* vol. 15, p. 6, Article ID: 14, 2021.

[5] B. A. Marzoog and T. I. Vlasova, "The possible puzzles of BCG vaccine in protection against COVID-19 infection," *Egyptian Journal of Bronchology,* vol. 15, p. 4, Article ID: 7, 2021.

[6] S. Dagher, A. Hilal-Alnaqbi, B. Akhozheya, and L. Akhu-Zaheya, "Utilizing Continuous Flow Microdevices-based Magnetophoresis for COVID-19 Diagnosis," *Biointerface Research in Applied Chemistry,* vol. 11, pp. 11996-12005, 2021.

[7] M. Azhar, R. Phutela, M. Kumar, A. H. Ansari, R. Rauthan, S. Gulati*, et al.*, "Rapid and accurate nucleobase detection using FnCas9 and its application in COVID-19 diagnosis," *Biosensors & Bioelectronics,* vol. 183, p. 8, Article ID: 113207, 2021.

[8] N. Zaki and E. A. Mohamed, "The estimations of the COVID-19 incubation period: A scoping reviews of the literature," *Journal of Infection and Public Health,* vol. 14, pp. 638-646, 2021.

[9] S.-H. Wang, "Covid-19 Classification by FGCNet with Deep Feature Fusion from Graph Convolutional Network and Convolutional Neural Network," *Information Fusion,* vol. 67, pp. 208-229, 2021.

[10] N. Maharjan, N. Thapa, B. P. Magar, M. Maharjan, and J. C. Tu, "COVID-19 Diagnosed by Real-Time Reverse Transcriptase-Polymerase Chain Reaction in Nasopharyngeal Specimens of Suspected Cases in a Tertiary Care Center: A Descriptive Cross-sectional Study," *Journal of Nepal Medical Association,* vol. 59, pp. 464-467, 2021.

[11] S. C. Satapathy, "A five-layer deep convolutional neural network with stochastic pooling for chest CT-based COVID-19 diagnosis," *Machine Vision and Applications,* vol. 32, Article ID: 14, 2021.

[12] X. Cheng, "PSSPNN: PatchShuffle Stochastic Pooling Neural Network for an Explainable Diagnosis of COVID-19 with Multiple-Way Data Augmentation," *Computational and Mathematical Methods in Medicine,* vol. 2021, Article ID: 6633755, 2021.

[13] X. Yao, "COVID-19 Detection via Wavelet Entropy and Biogeography-based Optimization," in *COVID-19: Prediction, Decision-Making, and its Impacts*, K. C. Santosh and A. Joshi, Eds., ed: Springer, 2020, pp. 69-76.

[14] Y. Chen, "Covid-19 Classification Based on Gray-Level Co-occurrence Matrix and Support Vector Machine," in *COVID-19: Prediction, Decision-Making, and its Impacts*, K. C. Santosh and A. Joshi, Eds., ed Singapore: Springer Singapore, 2020, pp. 47-55.

[15] Z. Lu, "A Pathological Brain Detection System Based on Radial Basis Function Neural Network," *Journal of Medical Imaging and Health Informatics,* vol. 6, pp. 1218-1222, 2016.

[16] S. Lu, "A Pathological Brain Detection System based on Extreme Learning Machine Optimized by Bat Algorithm," *CNS & Neurological Disorders - Drug Targets,* vol. 16, pp. 23-29, 2017.

[17] X. Wu, "Diagnosis of COVID-19 by Wavelet Renyi Entropy and Three-Segment Biogeography-Based Optimization," *International Journal of Computational Intelligence Systems,* vol. 13, pp. 1332-1344, 2020.

[18] S.-H. Wang, "COVID-19 classification by CCSHNet with deep fusion using transfer learning and discriminant correlation analysis," *Information Fusion,* vol. 68, pp. 131-148, 2021.

[19] W. Zhu, "ANC: Attention Network for COVID-19 Explainable Diagnosis Based on Convolutional Block Attention Module," *Computer Modeling in Engineering & Sciences,* vol. 127, pp. 1037-1058, 2021.

[20] K. S. Lee, J. Y. Kim, E. T. Jeon, W. S. Choi, N. H. Kim, and K. Y. Lee, "Evaluation of Scalability and Degree of Fine-Tuning of Deep Convolutional Neural Networks for COVID-19 Screening on Chest X-ray Images Using Explainable Deep-Learning Algorithm," *J Pers Med,* vol. 10, 2020.

[21] S. Minaee, R. Kafieh, M. Sonka, S. Yazdani, and G. Jamalipour Soufi, "Deep-COVID: Predicting COVID-19 from chest X-ray images using deep transfer learning," *Med Image Anal,* vol. 65, p. 101794, 2020.

[22] S. Toraman, T. B. Alakus, and I. Turkoglu, "Convolutional capsnet: A novel artificial neural network approach to detect COVID-19 disease from X-ray images using capsule networks," *Chaos Solitons Fractals,* vol. 140, p. 110122, 2020.

[23] L. Wang, Z. Q. Lin, and A. Wong, "COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images," *Sci Rep,* vol. 10, p. 19549, 2020.

[24] X. Yu, S. H. Wang, and Y. D. Zhang, "CGNet: A graph-knowledge embedded convolutional neural network for detection of pneumonia," *Inf Process Manag,* vol. 58, p. 102411, 2021.

[25] M. Polsinelli, L. Cinque, and G. Placidi, "A light CNN for detecting COVID-19 from CT scans of the chest," *Pattern Recognit Lett,* vol. 140, pp. 95-100, 2020.

[26] A. Shakarami, M. B. Menhaj, and H. Tarrah, "Diagnosing COVID-19 disease using an efficient CAD system," *Optik (Stuttg),* vol. 241, p. 167199, 2021.

[27] N. Iqbal, R. Mumtaz, U. Shafi, and S. M. H. Zaidi, "Gray level co-occurrence matrix (GLCM) texture based crop classification using low altitude remote sensing platforms," *Peerj Computer Science*, p. 26, Article ID: e536, 2021.

[28] D. H. Kim and S. Y. Ye, "Classification of Chronic Kidney Disease in Sonography Using the GLCM and Artificial Neural Network," *Diagnostics,* vol. 11, p. 12, Article ID: 864, 2021.

[29] S. Barburiceanu, R. Terebes, and S. Meza, "3D Texture Feature Extraction and Classification Using GLCM and LBP-Based Descriptors," *Applied Sciences-Basel,* vol. 11, p. 25, Article ID: 2332, 2021.

[30] S. Lu, "Detecting pathological brain via ResNet and randomized neural networks," *Heliyon,* vol. 6, p. e05625, 2020.

[31] A. Babaei, S. Banihashemi, and J. Damirchi, "Stable numerical solution of an inverse coefficient problem for a time fractional reaction-diffusion equation," *International Journal of Nonlinear Analysis and Applications,* vol. 12, pp. 365-383, 2021.

[32] S. C. Satapathy and D. Wu, "Improving ductal carcinoma in situ classification by convolutional neural network with exponential linear unit and rank-based weighted pooling," *Complex & Intelligent Systems,* vol. 7, pp. 1295-1310, 2021.

[33] Y.-D. Zhang, "Advances in Multimodality Data Fusion in Neuroimaging," *Information Fusion,* vol. 76, pp. 87-88, 2021.

[34] C. R. Luckett, R. Pellegrino, M. Heatherly, K. A. Martinez, M. Dein, and P. J. Munafo, "Discrimination of Complex Odor Mixtures: A Study Using Wine Aroma Models," *Chemical Senses,* vol. 46, p. 9, Article ID: bjaa079, 2021.

[35] D. N. Tan, M. Suvarna, Y. S. Tan, J. Li, and X. N. Wang, "A three-step machine learning framework for energy profiling, activity state prediction and production estimation in smart process manufacturing," *Applied Energy,* vol. 291, p. 14, Article ID: 116808, 2021.