Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

A self-supervised COVID-19 CT recognition system with multiple regularizations

Han Lu, Qun Dai [*]

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, PR China

ARTICLE INFO

Keywords:
Contrastive learning
COVID-19 CT Diagnosis
Loss regularization
Ensemble learning
Deep neural network

ABSTRACT

The diagnosis of Coronavirus Disease 2019 (COVID-19) exploiting machine learning algorithms based on chest computed tomography (CT) images has become an important technology. Though many excellent computer-aided methods leveraging CT images have been designed, they do not possess sufficiently high recognition accuracy. Besides, these methods entail vast amounts of training data, which might be difficult to be satisfied in some real-world applications. To address these two issues, this paper proposes a novel COVID-19 recognition system based on CT images, which has high recognition accuracy, while only requiring a small amount of training data. Specifically, the system possesses the following three improvements: 1) Data: a novel redesigned BCELoss that incorporates Label Smoothing, Focal Loss, and Label Weighting Regularization (LSFLLW-R) technique for optimizing the solution space and preventing overfitting, 2) Model: a backbone network processed by two-phase contrastive self-supervised learning for classifying multiple labels, and 3) Method: a decision-fusing ensemble learning method for getting a more stable system, with balanced metric values. Our proposed system is evaluated on the small-scale expanded COVID-CT dataset, achieving an accuracy of 94.3%, a precision of 94.1%, a recall (sensitivity) of 93.4%, an F1-score of 94.7%, and an Area Under the Curve (AUC) of 98.9%, for COVID-19 diagnosis, respectively. These experimental results verify that our system can not only identify pathological locations effectively, but also achieve better performance in terms of accuracy, generalizability, and stability, compared with several other state-of-the-art COVID-19 diagnosis methods.

1. Introduction

By April 10, 2022, the total number of confirmed cases of Coronavirus Disease 2019 (COVID-19) in the world had exceeded 400 million, of which the number of deaths had exceeded 6 million. The major obstacles in controlling the spreading of COVID-19 are the asymptomatic infection, slow detection speed, and high infectivity. Now, the main method of detecting COVID-19 is the Reverse Transcription-Polymerase Chain Reaction (RT-PCR) test. However, due to the slow detection speed, the shortage of diagnostic kits, and the sharp increase in the number of infected people, the detection efficiency becomes very low. In addition, the sensitivity of the RT-PCR testing kits is not high, which means that the efficiency of detection will be further compromised due to the false-negative problem.

At present, many automatic systems based on medical images using Artificial Intelligence (AI) technologies to detect COVID-19 have been developed. Given a medical image, a proposed model or system needs to correctly classify it, that is, to identify whether it has the pathological features of COVID-19. Many AI detection methods are developed based on X-rays images [1–5]. While previous studies have shown that chest Computed Tomography (CT) scans exhibit clear radiological features of COVID-19 patients. Besides, CT devices are very popular [6]. Therefore, utilizing CT images to diagnose COVID-19 is a feasible and promising solution. Some studies have developed related COVID-19 diagnostic methods on basis of CT images [7–17]. The research work conducted in Refs. [3,7,10,11] requires that, the employed datasets need to contain a large number of training images, which is generally difficult to be achieved, in practice. While the datasets utilized in Refs. [5,7,15] are not publicly available. These undisclosed data hinder the further AI research for COVID-19 detection using medical images. For addressing these two problems, in Ref. [16], He et al. develop the first publicly accessible small COVID-19 chest CT dataset containing 746 images, by extracting the CT images from over 760 preprints in medRxiv and bioRxiv. And they develop a complex but high-accuracy method to detect COVID-19. And in Ref. [8], joint learning and multi-task learning are used to identify COVID-19 by employing this dataset, however, with not ideal

[*] Corresponding author.
E-mail address: daiqun@nuaa.edu.cn (Q. Dai).

https://doi.org/10.1016/j.compbiomed.2022.106149
Received 16 June 2022; Received in revised form 24 August 2022; Accepted 24 September 2022
Available online 29 September 2022

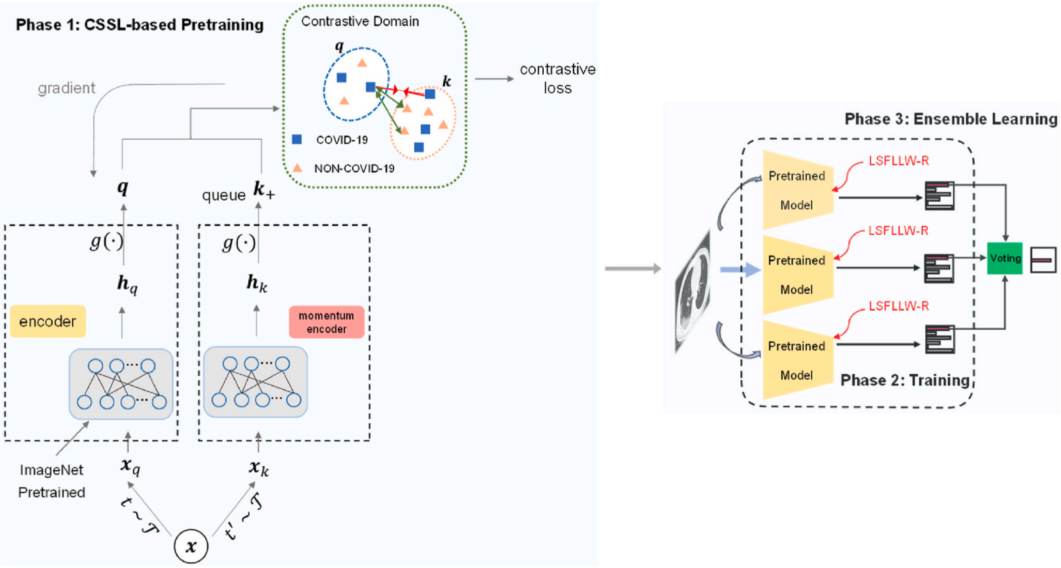


Fig. 1. The overview and pipeline of the proposed system, which includes three phases, i.e., Phase 1: CSSL-based Pretraining; Phase 2: Training; Phase 3: Ensemble Learning.

recognition accuracy. In Ref. [12], due to the expansion of the dataset, the single-label binary classification task of the previous researches has been transformed into six-label binary classification task. This new classification task has greatly accelerated the training speed, and through this task, using a simple neural network can achieve very excellent recognition performance. In Ref. [4], A patch-based CNN is proposed, which has fewer trainable parameters. This feature can make the network be trained stably under small datasets. Although these studies have made good progress, they still have great limitations. The core problem lies in that, these methods are unable to simultaneously take into account the amount of training data and recognition accuracy, that is, using limited training data to achieve superior recognition performance.

To address this challenge, we develop a novel deep ensemble learning system for COVID-19 detection based on chest CT scans, as shown in Fig. 1. The system is pretrained by taking advantage of the Contrastive Self-Supervised Learning (CSSL) paradigm [18], and multiple regularizations are employed within it to optimize the solution space during supervised training. First, due to the fact that the dataset is very small, we carefully design and propose a pretrained backbone network to replace the classical neural network architecture utilized in Ref. [12]. We pretrain the backbone network by performing CSSL on a big chest CT dataset without using labels. Then, we perform CSSL on the COVID-CT dataset, without using labels. This two-phase pretraining can bring good representational learning ability to a neural network. Moreover, the dataset proposed by Liu et al. in Ref. [12] is a multi-label binary classification dataset, and there exists extreme class imbalance in some labels. Consequently, on the basis of completing the CSSL-based pretraining, when we carry out the downstream supervised training tasks, aiming at these characteristics of the dataset, we redesign the BCELoss by integrating a novel Label Smoothing, Focal Loss, and Label Weighting Regularization (LSFLLW-R) technique proposed by us. We introduce Label Smoothing [19] to prevent overfitting, Focal Loss [20] (FL) to solve the extreme imbalance within some labels, and label weighting to pay more attention to the main task (COVID-19 binary classification) in the loss function. The regularizations fully exploit the characteristics of the dataset. Finally, we further employ the Bagging ensemble learning method to improve the generalizability of the system. Through the experimental verification on the expanded COVID-CT dataset, our proposed system outperforms several other state-of-the-art methods in terms of accuracy and other important metrics. Our main contributions in this work are summarized as follows:

- Considering the characteristics of the multi-label expanded COVID-CT dataset, we introduce a new cost-sensitive multiple regularizations technique LSFLLW-R, composed of Label Smoothing, Focal Loss, and label weighting, into BCELoss, which is more conducive to the identification of COVID-19.
- We employ the model pretrained by two-phase contrastive self-supervised learning as the backbone to facilitate the neural network to learn better representations.
- We utilize the Bagging ensemble learning algorithm to prevent overfitting and improve generalizability.
- We report the instability of the model trained without regularization on the expanded COVID-CT dataset, which can be alleviated by our proposed multiple regularizations technology LSFLLW-R.
- We perform extensive experiments to demonstrate the effectiveness of our proposed system. It achieves an accuracy of 0.943, a precision of 0.941, a recall (sensitivity) of 0.934, an F1-score of 0.947, and an Area Under the Curve (AUC) of 0.989 on the expanded COVID-CT dataset.

The rest of the paper is organized as follows. Section II reviews related works. Methodology and system framework are described in Section III. Section IV presents the experimental results and related analyses. In Section V, we conclude the paper and propose the prospect of future work.

2. Related work

The medical images such as CT and X-ray have played a great role in the struggle against COVID-19 [21]. And the fusion of AI technology and medical images further improves the power of the medical images. There are some excellent deep learning methods developed for the COVID-19 classification task using chest CT images. He et al. establish the first openly accessible COVID-19 chest CT dataset, i.e., COVID-CT, by extracting the CT images from over 760 preprints in medRxiv and bioRxiv and propose a deep learning method Self-Trans based on transfer learning and contrastive self-supervised learning [16]. Their method has achieved very good recognition accuracy (0.86) on this small CT dataset. However, this well-designed method needs many times of pretraining. Wang et al. propose a framework that performs joint learning of two datasets and performs multi-task learning consisting of classification task and supervised contrastive learning task to identify COVID-19 [8]. Their recognition accuracy (0.80) is not very high, but their method is cross-site. That is, in their study, they consider the different imaging conditions in the actual application scenarios. Liu et al. expand the dataset of [16] from 746 single-label CT images to six-label images [12]. They prove that these five additional labels can promote the training of main task. In addition, they also collect more CT images, so as to further improve the recognition accuracy. Their LA-DNN model greatly shortens the training time due to the addition of multiple auxiliary labels. Huang et al. point out that the method based on deep learning is difficult to deal with imprecise and uncertain information due to the low contrast of CT images, so they develop a classification network based on belief function using semi-supervised learning [9]. Ewen et al. propose a targeted self-supervised method, which makes the network architecture used by pretext tasks for self-supervision and downstream tasks unchanged, simplifying the experimental process, and enabling all layers of the network to gain benefits from self-supervised learning [17]. Mishra et al. combine the prediction results of several different deep CNN models to identify COVID-19 [13]. Among many competitive methods, this decision fusion method achieves the highest recognition accuracy (0.8834) on the COVID-CT dataset.

3. Method

Fig. 1 shows the overview of our established system for COVID-19 detection. In this section, we first introduce the redesigned BCELoss. In the supervised training phase, the redesigned BCELoss will be integrated with the regularization technology LSFLLW-R, and then, both of them will play their roles, synergistically. We then introduce the pretraining method implemented in the CSSL-based Pretraining phase. Finally, the algorithm of the whole proposed system is given.

3.1. BCELoss integrated with LSFLLW-R

In the course of our research, we find that the characteristics of the dataset are not taken into account in the research work carried out by Liu et al. in Ref. [12]. Inspired by this discovery, we construct a new approach by integrating the loss function BCELoss with a specially designed regularization technology LSFLLW-R, from the perspective of data distribution and prior knowledge, after deep consideration and analysis. The proposed multiple regularizations in the BCELoss improve the performance by reducing the solution space.

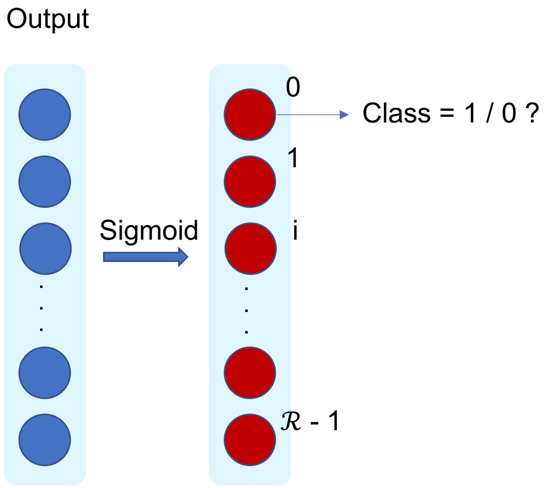


Fig. 2. $\mathcal{R}$-label binary classification.

1) Label Smoothing: As a regularization technique, Label Smoothing is proposed in Ref. [19]. The authors hope that the prediction of the model will not be too confident to generalize well. In other words, the assignment of a full probability to the ground-truth label by the model may cause overfitting. Therefore, in order to make the model less extreme, the authors propose a mechanism to change the ground-truth label distribution.

For a training example x in a K classification problem, considering that the ground-truth label of x equals t, its label distribution is:

$$q(k|x) = \delta_{k,t} = \begin{cases} 1, k = t \\ 0, k \neq t \end{cases} \tag{1}$$

where $k\,\varepsilon\,\{0, 1, \ldots, K-1\}$, $\delta_{k,t}$ is Dirac delta. Therefore, the Cross Entropy (CE) is:

$$CE = -\sum_{k=0}^{K-1} log(p_k) \bullet \delta_{k,t} = -\sum_{k=0}^{K-1} log(p_k) q(k) \tag{2}$$

$$p_k = p(k|x) = \frac{exp\ (z_k)}{\sum_{i=0}^{K-1} exp\ (z_i)} \tag{3}$$

where z_i are the unnormalized log-probabilities. Now, the authors use a new label distribution $q^{'}(k|x)$ instead of the original label distribution $q(k|x)$:

$$q^{'}(k|x) = (1-\varepsilon)\delta_{k,t} + \varepsilon u(k) \tag{4}$$

where $q^{'}(k|x)$ is a mixture of the original ground-truth distribution $q(k|x)$ and the fixed distribution $u(k)$, with weights $1-\varepsilon$ and ε, respectively.

In this paper, we follow the authors and use the uniform distribution $u(k) = \frac{1}{K}$, so that:

$$q^{'}(k) = (1-\varepsilon)\delta_{k,t} + \frac{\varepsilon}{K} \tag{5}$$

Thus, the CE has changed because of the change of label distribution. Note that, we use $k\,\varepsilon\,\{0, 1, \ldots, K-1\}$ instead of $k\,\varepsilon\,\{1, 2, \ldots, K\}$ used by the authors in Ref. [19], which is for the unification of the later formula.

2) Focal Loss: In the field of object detection, Lin et al. point out a fact that the object detectors with the highest accuracy are designed based on a two-stage approach popularized by R–CNN, and then a classifier is utilized to process a sparse set of candidate object locations. On the contrary, the one-stage object detectors with a regular and dense sampling of possible object locations may be faster and simpler, but they lag behind the two-stage detectors so far, in accuracy. Then, the authors discover that the extreme foreground-background class imbalance encountered during the training of dense detectors is the central reason [20].

For a training example x in a binary classification problem, considering that the ground-truth label of x equals t, the authors first introduce a weighting factor $\alpha \in [0, 1]$.

$$\alpha_t = \begin{cases} \alpha, t = 1 \\ 1-\alpha, t = 0 \end{cases} \tag{6}$$

So, the α-balanced CE loss is expressed as follows:

$$CE(p, t) = CE(p_t) = -\alpha_t\ log(p_t) \tag{7}$$

where

$$p_t = \begin{cases} p, t = 1 \\ 1-p, t = 0 \end{cases} \tag{8}$$

where $p \ \varepsilon \ [0,1]$ is the estimated probability for the class with label $t = 1$.

And then, the authors add a modulating factor $(1-p_t)^{\gamma}$ to the α-balanced CE loss, with a tunable *focusing* parameter $\gamma \geq 0$. And the α-balanced variant of the focal loss is expressed as follows:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma} \ log(p_t) \tag{9}$$

We can find that, for a training example $\boldsymbol{x}$, when $p_t \rightarrow 1$, $\boldsymbol{x}$ is an easy sample. Because according to Eq. (8), when $p_t \rightarrow 1$, $\boldsymbol{x}$ can be classified accurately. However, when $p_t \rightarrow 0$, $\boldsymbol{x}$ is misclassified, so $\boldsymbol{x}$ is a difficult sample. Therefore, after the modulating factor is added, the loss of well-classified samples goes to 0, and a large number of easy samples will not drown the classifier. In contrast, for hard samples, the modulation factor is near 1, the weights of these samples in the loss function will become larger than those of easy samples. In short, the *FL* function prevents the vast number of easy negatives from overwhelming the detector during training.

As a binary classification loss, BCELoss requires the network output to be processed by the Sigmoid function. Considering $\mathscr{R}$ labels, $i\varepsilon\{0,1,\ldots,\mathscr{R}-1\}$, $\mathscr{R}$-label binary classification can be observed from Fig. 2.

The loss of the i-th label is as follows:

$$L_i = -[t_i \ log(p_i) + (1-t_i)log(1-p_i)] \tag{10}$$

where p_i is the output result of i-th neuron processed by the Sigmoid function. And $t_i \ \varepsilon \ \{0,1\}$ is the ground-truth label value of i-th label. Aiming at the problem of class imbalance in some labels, now we introduce *FL* into the loss of i-th label:

$$L_i = -[\alpha(1-p_i)^{\gamma} t_i \ log(p_i) + (1-\alpha)p_i^{\gamma}(1-t_i)log(1-p_i)] \tag{11}$$

And, according to Eq. (5), the Label Smoothing in binary classification can be expressed as:

$$q^{'}(k) = (1-\varepsilon)\delta_{k,y} + \frac{\varepsilon}{K} = (1-\varepsilon)\delta_{k,y} + \frac{\varepsilon}{2} = \begin{cases} 1-\frac{\varepsilon}{2}, k = t_i \\ \frac{\varepsilon}{2}, k \neq t_i \end{cases} \tag{12}$$

where $k \in \{0,1\}$.

Therefore, in binary classification, the one-hot encoding is from $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ to $\begin{pmatrix} \frac{\varepsilon}{2} \\ 1-\frac{\varepsilon}{2} \end{pmatrix}$ or from $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ to $\begin{pmatrix} 1-\frac{\varepsilon}{2} \\ \frac{\varepsilon}{2} \end{pmatrix}$.

However, in BCELoss, we use a neuron for binary classification. Because the output of a single neuron is the probability for the class with label $t_i = 1$, we can convert by the following formula:

$$t_i = \begin{cases} 1-\frac{\varepsilon}{2}, t_i = 1 \\ \frac{\varepsilon}{2}, t_i = 0 \end{cases} \tag{13}$$

Now, we introduce Label Smoothing into the loss of i-th label as follows:

$$L_i = \begin{cases} -\left[\alpha(1-p_i)^{\gamma}\left(1-\frac{\varepsilon}{2}\right)log(p_i) + (1-\alpha)p_i^{\gamma}\left(\frac{\varepsilon}{2}\right)log(1-p_i)\right], t_i = 1 \\ -\left[\alpha(1-p_i)^{\gamma}\left(\frac{\varepsilon}{2}\right)log(p_i) + (1-\alpha)p_i^{\gamma}\left(1-\frac{\varepsilon}{2}\right)log(1-p_i)\right], t_i = 0 \end{cases} \tag{14}$$

Although the task is a six-label binary classification one, our main goal is to determine whether the subject has COVID-19 based on the chest CT images. Therefore, we give the highest weight to the main task (the COVID-19 recognition task), and the weights of the other five auxiliary tasks are directly proportional to their correlation with the main task. In Ref. [12], the authors gave the plots of the pairwise relationships among the five lesions on classifying COVID-19. As exhibited in the plots, the incidence of CrPa, AirBr, and InSepThi with COVID-19 are relatively high, that is to say, their potential correlation with COVID-19 may be stronger. Consequently, their weights should also be set relatively high. The redesigned BCELoss integrated with LSFLLW-R for the i-th label is as follows:

$$L_i = -w_i \begin{cases} \left[\alpha(1-p_i+\varphi)^{\gamma}\left(1-\frac{\varepsilon}{2}\right)log(p_i+\varphi) + (1-\alpha)(p_i+\varphi)^{\gamma}\left(\frac{\varepsilon}{2}\right)log(1-p_i+\varphi)\right], t_i = 1 \\ \left[\alpha(1-p_i+\varphi)^{\gamma}\backslash\left(\frac{\varepsilon}{2}\right)log(p_i+\varphi) + \left(1-\alpha\right)\left(p_i+\varphi^{)}\left(1-\frac{\varepsilon}{2}\right)log(1-p_i+\varphi)\right], t_i = 0 \end{cases} \tag{15}$$

where w_i is the weight of the i-th label, and constant $\varphi = 1e-5$ is introduced to let $log \ (\bullet) \neq \infty$. Now, we have obtained the loss of the i-th label of a single sample. Subsequently, we can average, sum or weight the losses of all labels as needed.

FL is considered from the point of view within the label, and label weighting is considered from the point of view between labels. Both of them are cost-sensitive strategies. As far as we know, this new approach combining multiple regularizations technique with BCELoss is innovatively constructed by us. By introducing LSFLLW-R into BCELoss, we achieve: 1) preventing a large number of negative samples in some labels from overwhelming the classifier, resulting in label failure and even poor impact on the network; 2) dividing the tasks into primary and secondary ones, and making the network more focused on the primary task, while the auxiliary tasks only served as facilitators, and 3) letting all labels be processed by Label Smoothing, and thus preventing overfitting. Accordingly, LSFLLW-R promotes the recognition accuracy by smoothing the loss function from the perspective of data.

3.2. An elaborately designed pretraining method

Because the target dataset (expanded COVID-CT dataset) is a small-scale dataset, pretraining should be implemented to prevent overfitting. As the two most mainstream paradigms of pretraining, self-supervised learning and transfer learning are widely used. Recently, Contrastive Self-Supervised Learning (CSSL) [18], as a dominant self-supervised learning method, has shown strong results in Natural Language Processing (NLP) and Computer Vision (CV), even beyond transfer learning. With transfer learning paradigm, labels of the source task are utilized for pretraining, making the pretrained model more inclined to the label distribution of the source task, and, eventually, resulting in poor generalization performance of the model in fulfilling the target task [22]. While the unsupervised pretraining implemented in CSSL can alleviate this problem, due to the fact that it only uses data instances to mine useful information. And CSSL has achieved good recognition results for the COVID-CT dataset [8,16,17]. In general, CSSL-based pretraining is less prone to overfitting. Based on the above analysis, we decide to conduct CSSL-based pretraining on the target dataset.

Contrastive Self-Supervised Learning: In CV, CSSL usually adopts the Siamese architecture [23]. For the past few years, many CSSL methods of good performance have been successfully formed. The MoCo v1 and v2 methods [24,25] introduce a queue to store negative samples, which decouples the dictionary size from the mini-batch size. The SimCLR method [26] achieves a simpler end-to-end contrastive loss mechanism by increasing the batch size, and by using the data augmentation and projection head techniques. The BYOL method [27] proposes a contrastive learning method without using negative samples. And the MoCo v3 and DINO methods [28,29] perform contrastive learning based on the Vision Transformer [30] model.

Augmenting a given image $\boldsymbol{x}$ to obtain the augmented images $\boldsymbol{x}_q$ and $\boldsymbol{x}_k$. $\boldsymbol{x}_q$ is the query and $\boldsymbol{x}_k$ is the key. Then, they are input into the encoder $f_q(\boldsymbol{x}_q;\theta_q)$ and $f_k(\boldsymbol{x}_k;\theta_k)$ parameterized by θ_q and θ_k, to obtain the representations $\boldsymbol{q}$ and $\boldsymbol{k}$. For N images $\{\boldsymbol{x}_i\}_{i=1}^{N}$, with N being the number of samples in the randomly sampled minibatch, we can obtain the representations $\{\boldsymbol{q}_i\}_{i=1}^{N}$ and $\{\boldsymbol{k}_i\}_{i=1}^{N}$. A positive pair is composed of a query image and a key one generated from the same image, and correspondingly, a negative pair is composed of two augmented images generated from different images. The idea of CSSL is to enlarge the similarity of positive pairs and reduce the similarity of negative pairs to enable the model to learn excellent representations. Therefore, the pretext task of CSSL can be an instance discrimination task, which judges whether a sample pair is positive or negative. Typical contrastive loss functions are InfoNCE [31] and NT-Xent [26].

For a representation $\boldsymbol{q}_j$, the InfoNCE loss function is as follows:

$$L = -log\frac{exp\ \left(\boldsymbol{q}_j \bullet \boldsymbol{k}_+/\tau\right)}{exp\left(\boldsymbol{q}_j \bullet \boldsymbol{k}_+/\tau\right) + \sum_{i=1}^{K} exp\ \left(\boldsymbol{q}_j \bullet \boldsymbol{k}_i/\tau\right)} \tag{16}$$

And the NT-Xent loss function is as follows:

$$L = -log\frac{exp\ \left(sim\left(\boldsymbol{q}_j, \boldsymbol{k}_+\right)\backslash/\tau\right)}{\sum_{i=1}^{N} I \bullet \left(exp\ \left(sim\left(\boldsymbol{q}_j, \boldsymbol{k}_i\right)/\tau\right) + exp\ \left(sim\left(\boldsymbol{q}_j, \boldsymbol{q}_i\right)\Big/\tau\right)\right)} \tag{17}$$

$$sim(\ \bullet\ ) = \frac{\boldsymbol{q}_j^T \bullet \boldsymbol{k}_+}{\left\|\boldsymbol{q}_j\right\| \bullet \left\|\boldsymbol{k}_+\right\|} \tag{18}$$

where τ is a temperature parameter, $\boldsymbol{k}_+$ and $\boldsymbol{q}$ constitute the positive pair, $I \in \{0,1\}$ is an indicator factor evaluating to 1 if $j \neq i$, K refers to the number of negative samples, and Eq. (18) is the cosine similarity, which is used to measure the similarity of representations.

Our CSSL-based pretraining process is illustrated in Phase 1 of Fig. 1, from which it can be clearly perceived that, a novel contrastive loss mechanism [24,25] is adopted, neither end-to-end nor memory bank. For the end-to-end mechanism, although it can update the encoders of query and key at the same time to maintain consistency, the dictionary size cannot be very large due to the limitation of GPU memory size. For the memory bank mechanism, it can support a large dictionary size and make the key encoder consistent. However, only updating the encoder of the query will make the query and key less consistent. Comprehensively considering the above analyses, we use the on-the-fly queue structure to store keys, so as to decouple the dictionary size from the batch size, and to obtain a larger dictionary size. Accordingly, in our COVID-19 classification research work, we can provide more negative examples to participate in contrastive learning, so as to facilitate convergence. Naturally, we use the InfoNCE loss, with reference to the MoCo method. At the same time, we perform the momentum update on both of the two encoders, to make the encoder of query and key, and the encoder of key and key consistent. Formally, the two encoders are denoted as $f_q(\boldsymbol{x}_q;\theta_q)$ and $f_k(\boldsymbol{x}_k;\theta_k)$, being parameterized by θ_q and θ_k. The specific momentum update rule is as follows:

$$\theta_q \leftarrow \theta_q - \alpha\frac{\partial\mathscr{L}}{\partial\theta_q} \tag{19}$$

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q \tag{20}$$

where $m\ \varepsilon\ [0,1)$ is a momentum coefficient and α is the learning rate.

However, only implementing pretraining based upon the target dataset may not be sufficient. For injecting more knowledge into the model, similar to the research work conducted in Ref. [16], before pretraining on the target dataset, we pretrain the model on basis of a big Lung Nodule Analysis (LUNA) dataset[1], by using the same pretraining method. The learning paradigm implemented in this stage can be regarded as either transfer learning or self-supervised learning. In this paper, we adopt the term self-supervised learning rather than transfer learning. Through the CSSL-based pretraining on the large-scale dataset, the model can acquire feature representations that are not biased towards labels of related source domain datasets. All in all, we provide the model pretrained by two-phase CSSL for the downstream supervised tasks. By means of the instance discrimination task, the feature space is constrained, and the model can further achieve intra-class cohesion and inter-class separation.

And we introduce the Multilayer Perceptron (MLP) projection head proposed in SimCLR by Chen et al. of [26], that is, the embedded network $g(\bullet)$ in Fig. 1. By projecting $\boldsymbol{h}_q$ and $\boldsymbol{h}_k$ into low-dimensional space, $\boldsymbol{h}_q$ and $\boldsymbol{h}_k$ can form and maintain more information, that is, form a qualitative representation. Experiments show that the pretrained base models with unbiased feature representations is more conducive to further ensemble. Lastly, the results of the experiments conducted in Refs. [16,17] show that, DenseNet-169 [32] is more suitable for the COVID-CT dataset than the other convolutional neural network models. Therefore, we use the DenseNet-169 model pretrained on the ImageNet [33] dataset as the initialized model.

Through feature reuse, a more compact model DenseNet is acquired, an efficient neural network, which alleviates the problem of vanishing-gradient and greatly reduces the number of parameters. The general structure of DenseNet is displayed in Fig. 3. The dense connectivity block is the most important structure in DenseNet. Within the block, the number of feature maps becomes more and more through multiple non-linear transformation and concatenation. For the i-th layer in the block, its output u_i is obtained by the non-linear transformation after concatenating all the feature maps of the previous layer:

$$u_i = H_i([u_0, u_1, u_2, \ldots, u_{i-1}])$$

where $H_i(\bullet)$ refers to the non-linear transformation of layer i and $[u_0, u_1, u_2, \ldots, u_{i-1}]$ is the concatenation of all the feature maps produced from layers 0, 1, …, and i-1.

According to the different parameter settings, DenseNet can be distinguished as DenseNet-121, DenseNet-169, DenseNet-201 and DenseNet-161. In the DenseNet-169 model used in this paper, the first step is 7 × 7 convolution and 3 × 3 max pooling. Then, the non-linear transformation $H_i(\bullet)$ in each block can be designed as six continuous operations: BN-ReLU-Conv with kernel size 1 × 1 followed by BN-ReLU-Conv with kernel size 3 × 3. The numbers of layers of the four blocks are 6, 12, 32, and 32, respectively. Within each block, the output of each layer contains 32 channel feature maps. And the connecting part Transition between two blocks is for down-sampling, which consists of 1 × 1 convolution, followed immediately by 2 × 2 average pooling.

To summarize, based on the DenseNet-169 model pretrained on the ImageNet dataset, we further perform CSSL-based pretraining on the large LUNA dataset, and then perform CSSL on the target dataset COVID-CT. In this way, the final pretrained model is obtained. It is worth noting that, the two-phase pretraining belongs to self-supervised pretraining, implying that dataset labels are not required.

3.3. Overall system framework design

The specific design and implementation details of our proposed system are presented in Algorithm 1, which corresponds to Fig. 1, precisely. Specifically, for a given input image $\boldsymbol{x}$, we use the data augment of same distribution to form two images, and then carry out two-phase contrastive learning is the first phase. The second phase is supervised

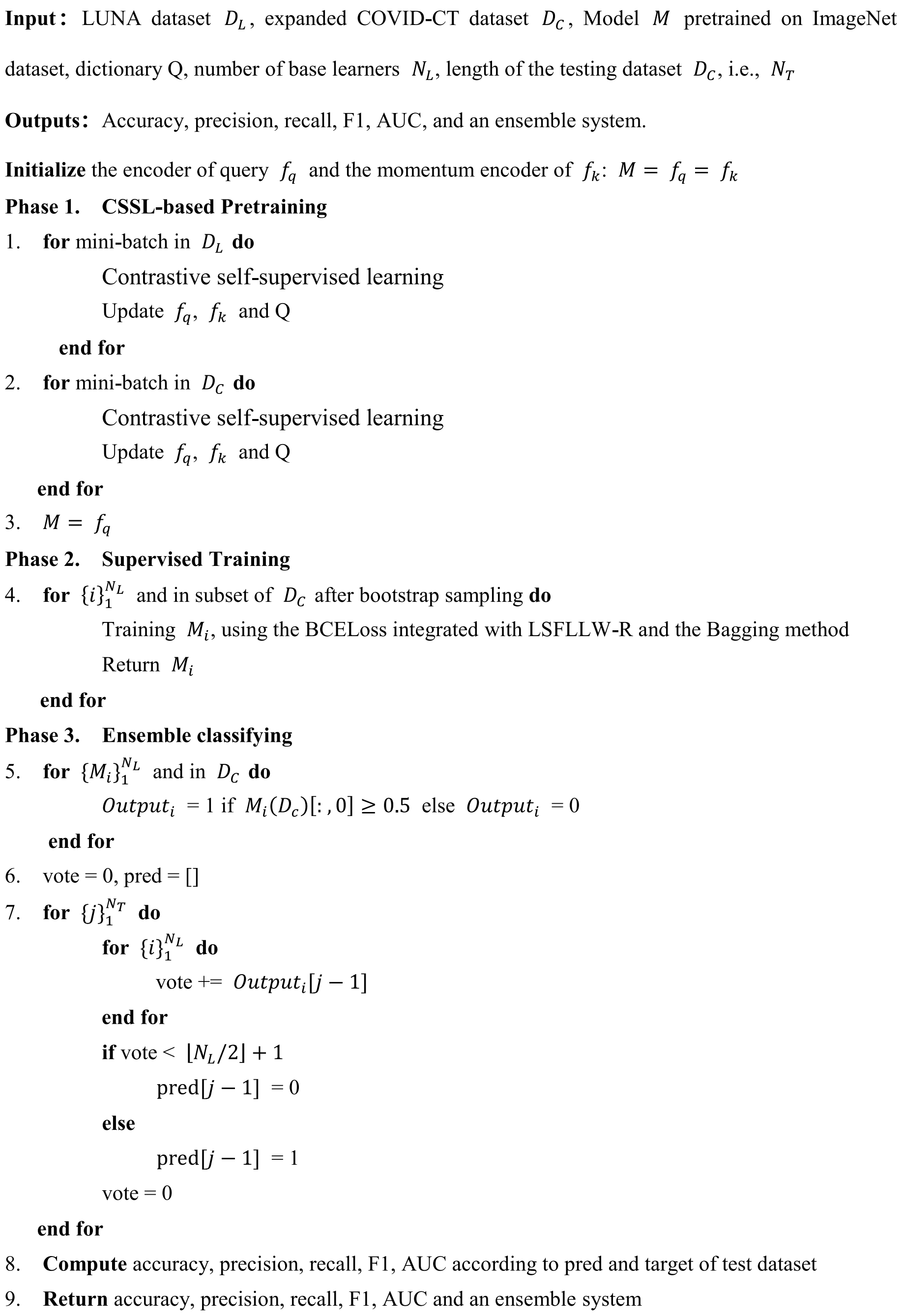
[1] https://luna16.grand-challenge.org/Data/.

training with the LSFLLW-R and ensemble learning techniques. The third phase is to adopt the plurality voting method for the classification results of multiple learners generated by ensemble learning.

Ensemble Learning: The ensemble learning paradigm is generally constituted of two stages: 1) the generation of the individual base learners, and 2) the combination of these individual base learners, if the intermediate ensemble pruning stage is not taken into consideration here. An ensemble system can be composed of several homogeneous or heterogeneous member models, also known as individual base learners, as mentioned above. Homogeneous models are produced from multiple different executions of the same learning algorithm. These homogeneous models can be generated by setting different parameter values of the learning algorithm, introducing random factors into the learning algorithm, or by manipulating training samples, attribute values of input variables and outputs of the model [34]. The most popular methods for generating homogeneous models are Bagging [35] and Boosting [36]. Heterogeneous models are produced by running different learning algorithms on the same dataset. Such heterogeneous models have different views about the data, because they hold different assumptions about the data.

There are usually three combination strategies of the member models in an ensemble: the averaging method, the voting method, and the learning method. The averaging method is usually employed used for regression tasks, and the voting method is usually used for classification tasks. The learning method is a more powerful combination strategy. Stacking [37] is a classic representative of the learning method. Ensemble learning integrates the decisions of multiple member models, which can usually achieve obtain better generalization performance than a single learner, and can effectively avoid overfitting.

Algorithm1. COVID-19 Recognition System



```
Input: LUNA dataset $D_L$, expanded COVID-CT dataset $D_C$, Model $M$ pretrained on ImageNet
dataset, dictionary Q, number of base learners $N_L$, length of the testing dataset $D_C$, i.e., $N_T$
Outputs: Accuracy, precision, recall, F1, AUC, and an ensemble system.
Initialize the encoder of query $f_q$ and the momentum encoder of $f_k$: $M = f_q = f_k$
Phase 1.  CSSL-based Pretraining
1.  for mini-batch in $D_L$ do
        Contrastive self-supervised learning
        Update $f_q$, $f_k$ and Q
    end for
2.  for mini-batch in $D_C$ do
        Contrastive self-supervised learning
        Update $f_q$, $f_k$ and Q
    end for
3.  $M = f_q$
Phase 2.  Supervised Training
4.  for $\{i\}_1^{N_L}$ and in subset of $D_C$ after bootstrap sampling do
        Training $M_i$, using the BCELoss integrated with LSFLLW-R and the Bagging method
        Return $M_i$
    end for
Phase 3.  Ensemble classifying
5.  for $\{M_i\}_1^{N_L}$ and in $D_C$ do
        $Output_i$ = 1 if $M_i(D_c)[:,0] \geq 0.5$ else $Output_i$ = 0
    end for
6.  vote = 0, pred = []
7.  for $\{j\}_1^{N_T}$ do
        for $\{i\}_1^{N_L}$ do
            vote += $Output_i[j-1]$
        end for
        if vote < $\lfloor N_L/2 \rfloor + 1$
            pred$[j-1]$ = 0
        else
            pred$[j-1]$ = 1
        vote = 0
    end for
8.  Compute accuracy, precision, recall, F1, AUC according to pred and target of test dataset
9.  Return accuracy, precision, recall, F1, AUC and an ensemble system
```

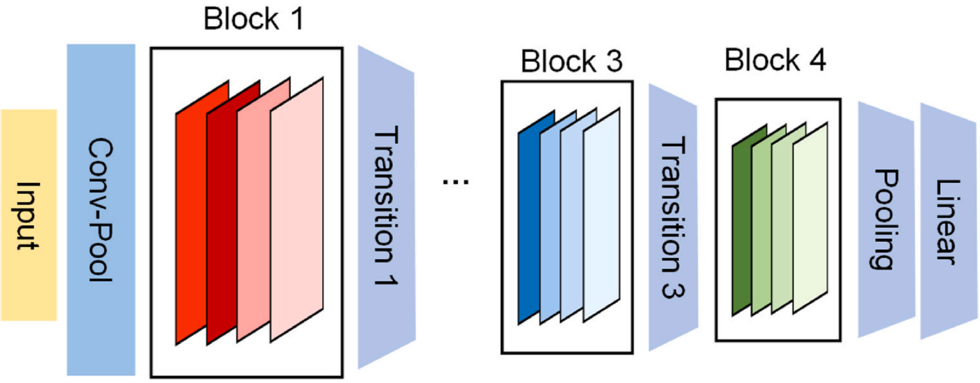


Fig. 3. The overview of deep DenseNet framework.

The plurality voting method, a type of voting method, makes an unweighted voting on the outputs of the member models in an ensemble. Assuming $\boldsymbol{x}$ represents an image input to the ensemble system, $F(\boldsymbol{x})$ is the output of the ensemble by utilizing the plurality voting method. The following Eq. (21) calculates $F(\boldsymbol{x})$, when the plurality voting method is implemented:

$$F(\boldsymbol{x}) = \underset{y}{argmax} \sum_{i=1}^{N_L} I(h_i(\boldsymbol{x}) = y), y \in Y \tag{21}$$

where $h_i(\boldsymbol{x})$ is the classification decision of the i-th model, and $i = \{1,2, \ldots, N_L\}$. N_L denotes

the number of generated base learners. $I(\bullet)$ represents the indicator function ($I(false) = 0, I(true) = 1$). And $Y = \{0, 1, \ldots, K-1\}$ is the set of class labels.

In Phase 3 of Fig. 1, we employ the simple but efficient Bagging algorithm for ensemble learning. Bagging, also known as bootstrap ensemble learning, is a method of repeatedly sampling from the original sample set with replacement, according to uniform probability distribution. The Bagging method reduces the generalization error of the ensemble system by reducing the variance of the base learners. Bagging does not focus on processing any particular instances of the training data. With the Bagging method, each sample has the same probability of being selected. Therefore, when Bagging is applied to noisy data, it is not susceptible to model overfitting [38].

According to the principle of Bagging algorithm, on the basis of completing the CSSL-based pretraining, we randomly take some samples from the original training dataset at a certain rate each time to form a subset of training samples, and then, multiple base models are produced by implementing supervised training on these training subsets, using the BCELoss loss function integrated with LSFLLW-R. Finally, the plurality voting method is utilized to determine the final COVID-19 recognition results in accordance with the predictive results of all the base models.

3.3.1. Datasets and metrics

In this study, all the datasets involved are composed of chest CT images.

a) A Part of the LUNA Dataset D_L: In the first stage of CSSL-based pretraining, we leverage the big Lung Nodule Analysis (LUNA) database. The LUNA dataset is developed for LUNA16 challenge, which contains 888 CT scans. In order to be consistent with the study conducted in Ref. [16], we use the same 1000 CT images selected from the LUNA dataset by the authors of [16]. Note that, during pretraining, we do not use the labels of these 1000 images.

b) Exp-COVID-CT [12] [2] D_C: The training and testing processes of our proposed system are performed based upon this dataset. The dataset is an expansion of the COVID-CT dataset proposed by He et al. in Ref. [16].[3] Therefore, the expanded dataset is abbreviated as Exp-COVID-CT, while the original COVID-CT dataset in Ref. [16] is abbreviated as Ori-COVID-CT, by us. The Ori-COVID-CT dataset consists of 349 COVID-19 CTs from 216 patients and 397 Non-COVID-19 CTs. Table 1 details this dataset. It is a single-label dataset, that is, it only contains category information about whether or not the subjects corresponding to the CT images have COVID-19. Liu et al. point out that the radiological reports of the COVID-19 positive images are of great value [12]. After a comprehensive statistical analysis of the entire text annotations, they found that there are five different lesions associated with COVID-19, including Ground Glass Opacity (GGO), Consolidation (Csld), Crazy Paving appearance (CrPa), Air Bronchograms (AirBr), and Interlobular Septal Thickening (InSepThi). Therefore, they expand the single-label dataset to a six-label dataset. In addition to the original single main label, five additional labels corresponding to the five lesions mentioned above are expanded into the dataset. Obviously, only when the COVID-19 label of a CT image is 1, its additional five labels can be taken as 1. We count the number of samples with a label value of 1 (corresponding to positive cases) in each auxiliary label of the training dataset with the size of 425, which is shown in Fig. 4. It is worth emphasizing that: a) expanding the five additional labels do not result in a change in the number of images in the dataset. Therefore, datasets Exp-COVID-CT and Ori-COVID-CT have the same image, and b) these additional labels are only intended to aid in the training. Therefore, datasets Exp-COVID-CT and Ori-COVID-CT are equivalent when testing. And the Ori-COVID-CT dataset will be used in our later comparative experiments. And

Table 1
Dataset split.

	# images			# patients	
	COVID	Non-COVID	All	COVID	Non-COVID
Train	191	234	425	130	105
Val	60	58	118	32	24
Test	98	105	203	54	42
All	349	397		216	171

[2] https://github.com/xiaoxuegao499/LA-DNN-for-COVID-19-diagnosis.

[3] https://github.com/UCSD-AI4H/COVID-CT.

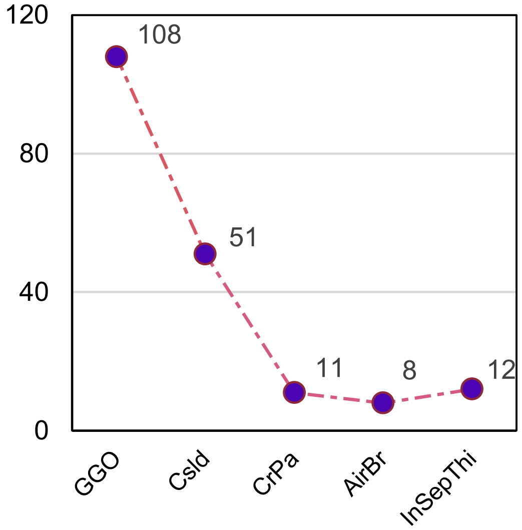


Fig. 4. The number of samples with a label value of 1 (corresponding to positive cases) in each auxiliary label of the training dataset with the size of 425.

Fig. 5 shows some examples of the LUNA, Ori-COVID-CT and Exp-COVID-CT datasets.

c) Exp-COVID-CT* [12]: In order to achieve higher recognition accuracy, Liu et al. continue to collect more CT images to form a larger dataset, which we call the dataset Exp-COVID-CT*. It contains 564 COVID-19 CTs and 660 Non-COVID-19 CTs. It is also a six-label dataset.

To compare various algorithms, we adopt five classical metrics, including accuracy (ACC), F1-score (F1), recall (REC), precision (PRE), and the Area Under the receiver operating characteristic Curve (AUC). Here, the ACC, F1, REC, and PRE are defined as,

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

$$F1 = 2 \bullet \frac{precision \bullet recall}{precision + recall} \tag{23}$$

$$REC = \frac{TP}{TP + FN} \tag{24}$$

$$PRE = \frac{TP}{TP + FP} \tag{25}$$

where TP, TN, FP, and FN represent True Positive, True Negative, False Positive, and False Negative, respectively.

4. Experiments

4.1. Experiment setup

The whole system is implemented in PyTorch. In the CSSL-based pretraining phase, our experiments are carried out with 4 Nvidia GTX 1080Ti GPUs using data parallelism. For the supervised training with LSFLLW-R and the ablation experiments, we use a desktop equipped with Intel Xeon E5-2670 2.6 GHz CPU and 64 GB memory. Then, we carefully design LSFLLW-R. We set $\varepsilon = 0.1$, $\gamma = 5$, $\alpha = 0.5$, and *weight list* $= [3.5 \times 5, 1.5, 1.5, 2.5, 2.5, 2.5]$. The imbalance of some auxiliary labels can be reflected in Fig. 4, apparently, the main label and labels GGO and Csld are relatively balanced, therefore, we only apply

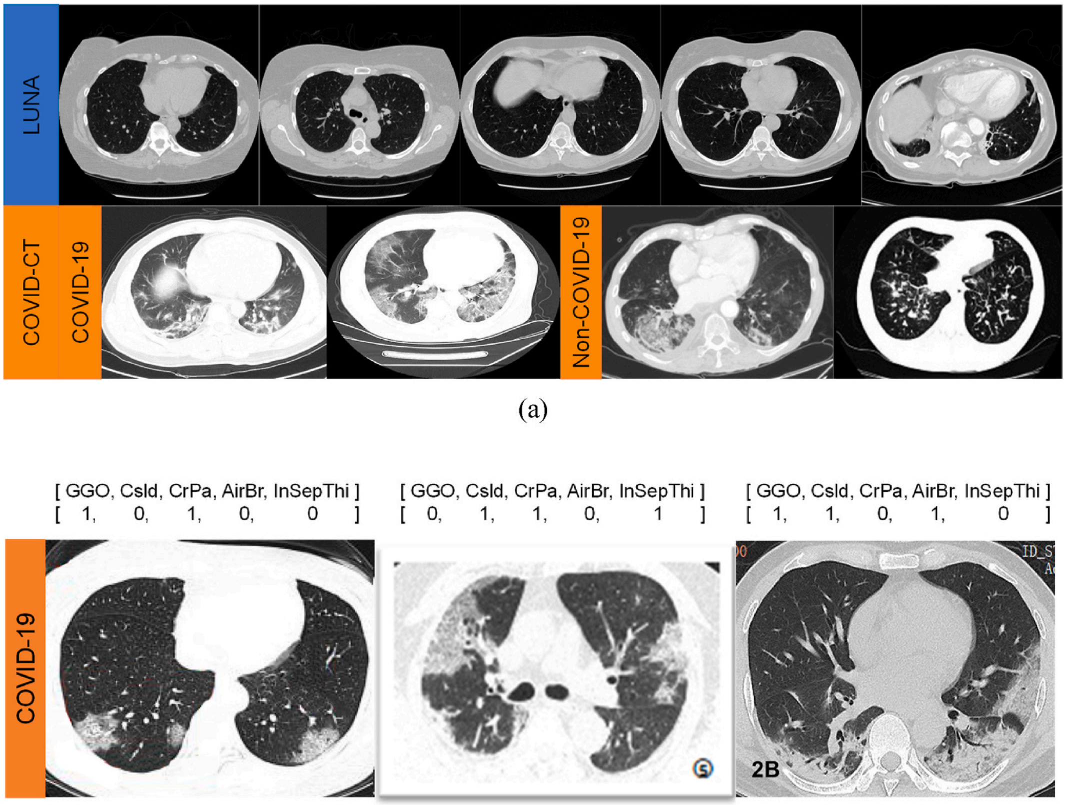

LUNA
COVID-CT
COVID-19
Non-COVID-19
(a)
[GGO, Csld, CrPa, AirBr, InSepThi]
[1, 0, 1, 0, 0]
[GGO, Csld, CrPa, AirBr, InSepThi]
[0, 1, 1, 0, 1]
[GGO, Csld, CrPa, AirBr, InSepThi]
[1, 1, 0, 1, 0]
COVID-19
2B
(b)

Fig. 5. (a) Examples of unlabeled LUNA dataset (upper), two positive CT scans for COVID-19 and two negative CT scans for COVID-19 of the COVID-CT dataset (lower); (b) Examples of the positive CT scans for COVID-19 of the COVID-CT dataset and its corresponding five auxiliary labels.

Table 2
Results for COVID-19 classification of different methods on three datasets.

Methods		ACC	F1	REC	PRE	AUC
In Ori-COVID-CT	Self-Trans [16]	0.86	0.85	0.79	0.92	0.94
	Wang et al. [8]	0.80	0.80	0.81	0.79	0.86
	Evidential Covid-Net [9]	0.81	0.812	\	\	0.875
	Mishra et al. [13]	0.8834	0.867	\	\	0.8832
	Ewen et al. [17]	0.8621	0.8704	\	\	0.8609
In Exp-COVID-CT	LA-DNN [12]	0.852	0.848	0.857	\	0.912
In Exp-COVID-CT*	LA-DNN [12]	0.877	0.868	0.874	\	0.933
Our method		**0.943**	**0.947**	**0.934**	**0.941**	**0.989**

the FL of LSFLLW-R to the CrPa, AirBr, and InSepThi. According to the preliminary experiment, we set the number of base learners for ensemble learning $N_L = 3$. Intuitively, this can be roughly explained as: our main task is binary classification and the regularized model is relatively stable, therefore, when $N_L = 3$, the wrong samples can be discarded by plurality voting. When $N_L > 3$, too many relatively stable models will be unprofitable, and even they might overwhelm the voting results and degrade the ensemble performance. We set the bootstrap sampling times $m = 256$. That is, the sampling rate is about 60.2%. Our method is repeated nine times, i.e., nine base models are trained. Then we select three models each time in sequence as an ensemble system, and finally, seven ensemble systems are obtained. All reported metric results are averages over these seven systems.

4.2. Comparison with state-of-the-art methods

Our experiments are based on the Exp-COVID-CT dataset proposed by Liu et al. in Ref. [12]. So, we directly compare our proposed method with the LA-DNN model in Ref. [12]. However, the research work conducted in Ref. [12] is pioneering, with no further well-designed method. And furtherly, a) as described in the related work, many researchers propose competitive state-of-the-art methods on the single-label Ori-COVID-CT dataset without additional disease labels, therefore, we also compare the results of our method on the Exp-COVID-CT dataset with the results of some state-of-the-art methods on the Ori-COVID-CT dataset, and b) we also compare the results of our method on the Exp-COVID-CT dataset with the results of LA-DNN on the Exp-COVID-CT* dataset with a large number of training images.

Table 2 shows the experimental results of different methods. The REC and PRE of Self-Trans are not given in Ref. [16], but we find them on their GitHub. As shown in Table 2, We can make the following comparison and draw relevant conclusions:

1) Compared with the LA-DNN on the Exp-COVID-CT dataset. On the Exp-COVID-CT dataset, as can be seen, the ACC of our proposed method is 0.943, the F1 is 0.947, the REC is 0.934, the PRE is 0.941 and the AUC is 0.989. We achieve a huge improvement of 9.1% in ACC, 9.9% in F1, 7.7% in REC, and 7.7% in AUC.
2) Compared with the LA-DNN on the Exp-COVID-CT* dataset. Our proposed method is implemented on the small Exp-COVID-CT dataset, while the results of LA-DNN on the larger Exp-COVID-CT* dataset are worse. This indicates that our proposed method is able to achieve competitive classification performances on the small-scale dataset.
3) Compared with several state-of-the-art methods on the Ori-COVID-CT dataset. We compare our method implemented on the small Exp-COVID-CT dataset with five state-of-the-art baseline methods implemented on the Ori-COVID-CT dataset. We achieve 8.3%, 14.3%, 13.3%, 5.96%, and 8.09% improvements in terms of ACC over the Self-Trans, the method proposed by Wang et al., Evidential Covid-Net, the method proposed by Mishra et al., and the method proposed by Ewen et al., respectively.

The highest mean accuracy is observed for our method with 0.943. The above promising results can reveal that our design is effective. We further give the comparison diagram between our method and several competitive methods on the Ori-COVID-CT dataset. See Fig. 6.

4.3. Evaluation of our method with ablation experiments

In order to further evaluate and understand our proposed system, we

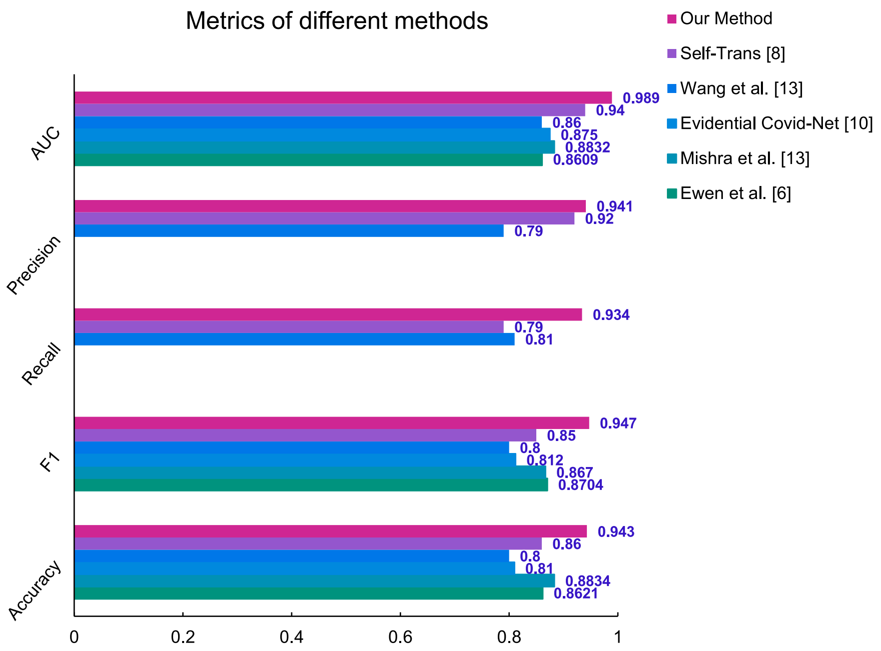


Fig. 6. The classification performance of our proposed method and a variety of competitive methods.

Table 3
Results of our proposed method in the ablation experiments.

	ACC	F1	REC	PRE	AUC
Sub-Method 1	0.805	0.789	0.760	0.826	0.878
Sub-Method 2	0.914	0.910	0.895	0.930	0.977
Sub-Method 3	0.938	0.934	0.914	**0.956**	0.982
Our method	**0.943**	**0.947**	**0.934**	0.941	**0.989**

Table 4
Pairwise accuracy t-test results between our method and the SUB-METHODS.

Methods	Sub-Method 1	Sub-Method 2	Sub-Method 3
p|H	1.8345e-11|1	0.0250|1	0.1479|0

Note: H value of 1 indicates that the classification performance of our proposed algorithm is significantly superior to other methods when the *t*-test is conducted pairwise base on accuracy at a 5% significance level.

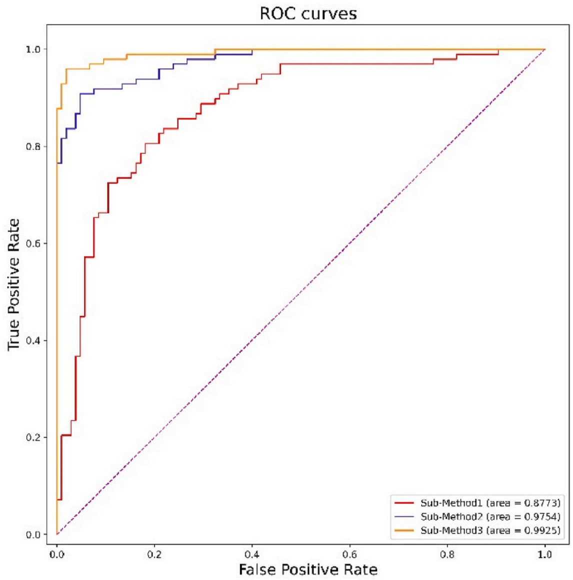


Fig. 7. ROC curves of three Sub-Methods in the ablation experiments.

conduct the following ablation experiments. The ablation experiments are divided into four groups of experiments by using the below three Sub-Methods and our proposed method, respectively.

- **Sub-Method 1:** Pretrain on the ImageNet dataset. Then, fine-tune on the Exp-COVID-CT dataset using labels. The LSFLLW-R and the ensemble learning techniques are not utilized.
- **Sub-Method 2:** Pretrain on the ImageNet dataset. Perform two-phase CSSL-based pretraining: apply CSSL on the LUNA dataset, and then on the Ori-COVID-CT dataset (Since the labels are not used in the pretraining stage, the Exp-COVID-CT dataset is equal to the COVID-CT dataset). Then, fine-tune on the Exp-COVID-CT dataset using labels. The LSFLLW-R and the ensemble learning techniques are not utilized.
- **Sub-Method 3:** Pretrain on the ImageNet dataset. Perform two-phase CSSL-based pretraining: apply CSSL on the LUNA dataset and then on the Ori-COVID-CT dataset. Then, fine-tune on the Exp-COVID-CT dataset using labels and the LSFLLW-R technique, but not the ensemble learning technique.
- **Our method:** Pretrain on the ImageNet dataset. Perform two-phase CSSL-based pretraining: apply CSSL on the LUNA dataset and then on the Ori-COVID-CT dataset. Then, fine-tune on the Exp-COVID-CT dataset using labels, and utilizing both the LSFLLW-R and the ensemble learning techniques.

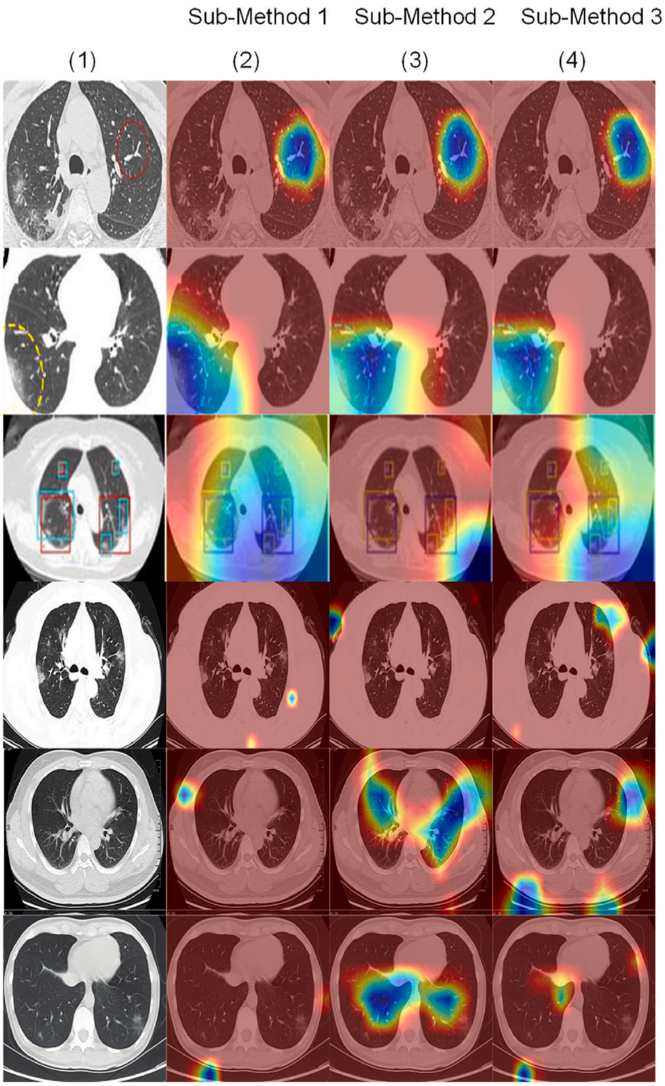


Fig. 8. Grad-CAM visualizations of the three Sub-Methods in the six positive CT images for COVID-19. The images in Column (1) are the original data and those in Columns (2–4) correspond to the visualization results of the three Sub-Methods implemented in our ablation experiments.

For Sub-Method 1 to Sub-Method 3, we conduct nine repeated experiments and report the average of each metric. For our method, experimental settings have been described in Section IV-A.

The corresponding results of the ablation experiment are presented in Table 3. By comparing the results (0.805, 0.789, 0.760, 0.826, and 0.878 for ACC, F1, REC, PRE, and AUC, respectively) of Sub-Method 1 with the results of those state-of-the-art methods implemented on the single-label Ori-COVID-CT dataset in Table 2, we can observe that our proposed method better than the method proposed by Wang et al. [8], which indicates that the additional labels of the Exp-COVID-CT dataset are helpful for training. However, Sub-Method 1 is still inferior to the other methods. This implies that the improvement of the results due to the introduction of multiple labels is limited. The model will encounter bottlenecks.

As shown in Table 3, some important observations can be summarized. First, by comparing the results of Sub-Method 2 with Sub-Method 1, the two-phase pretraining helps the model improve ACC by 10.9%, F1 by 12.1%, REC by 13.5%, PRE by 10.4%, and AUC by 9.9%. This

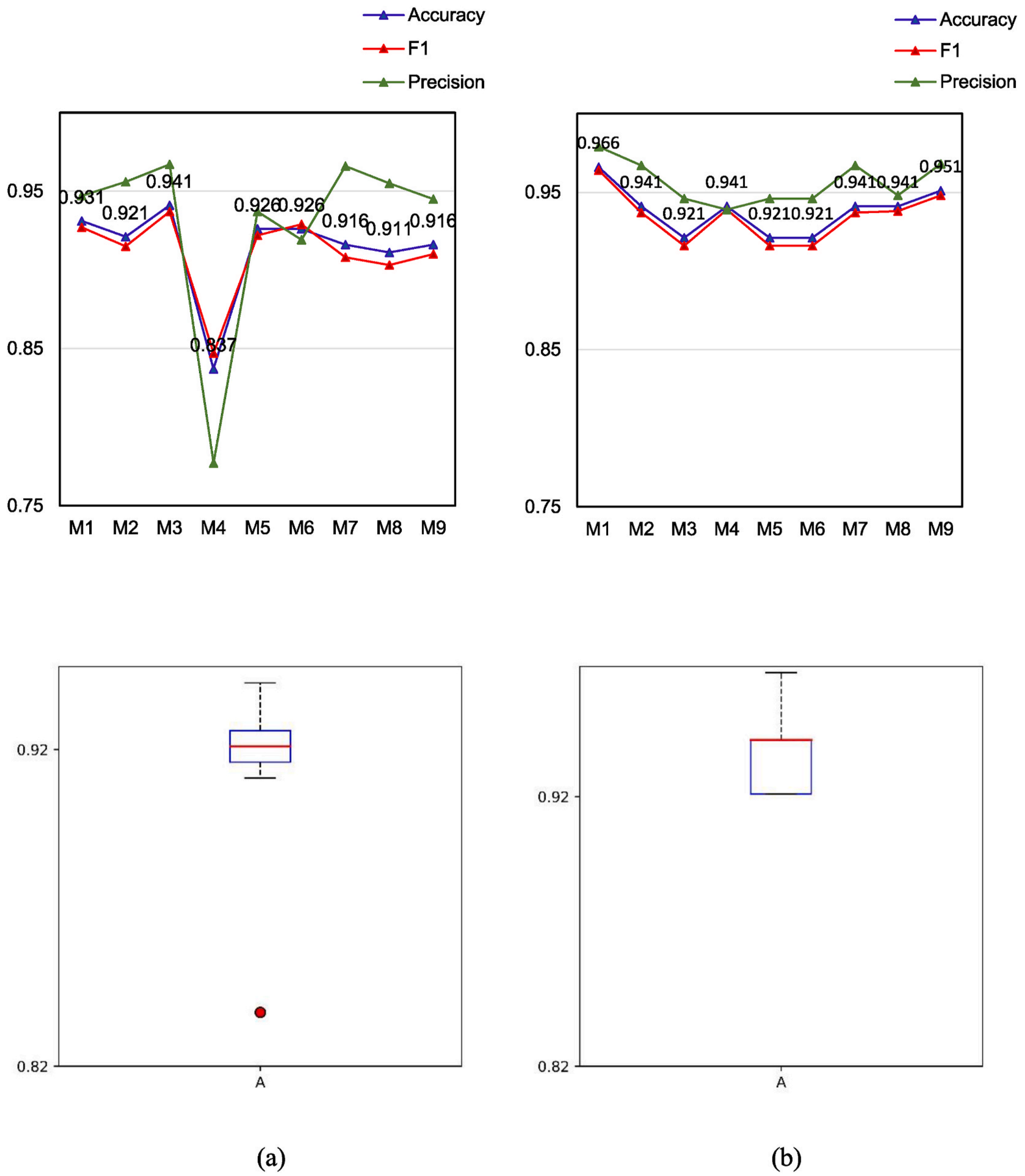


Fig. 9. (a) Three metrics of nine models trained by Sub-Method 2 (upper) and corresponding box plot of the ACC (lower); (b) Three metrics of nine models trained by Sub-Method 3 (upper) and corresponding box plot of the ACC (lower).

observation can illustrate that, after CSSL, the model has excellent representational learning ability. Unbiased feature representations have rich semantic information, which greatly improves the model. Second, the Sub-Method 3 outperforms Sub-Method 2. The slight improvement reveals the superiority of LSFLLW-R. It imposes constraints through multiple regularizations and overcomes the unfavorable factors caused by data characteristics in the training process. Thus, the model can identify more difficult samples. To this end, through further ensemble learning, we achieve 0.5%, 1.3%, 2%, and 0.7% better than Sub-Method 3 in average ACC, F1, REC, and AUC. But the PRE (0.941) yielded by our method is 1.5% worse than the Sub-Methods 3. We believe that, first of all, our ensemble has made progress in the four metrics, which is noteworthy. Additionally, the PRE of the three Sub-Methods is significantly higher than REC, which is an unbalanced performance. Therefore, although our PRE is not as good as Sub-Method 3, the results achieved by our method are more balanced across metrics (0.943, 0.947, 0.934, 0.941, and 0.989 for ACC, F1, REC, PRE, and AUC, respectively), that is to say, through decision fusion, the model can be applied better in all cases and has better robustness. At the same time, in order to further prove the effectiveness of our proposed method, we implement the *t*-test significance test. Specifically, we conduct t-tests pairwise between our method and Sub-Methods at the significance level of 5%. The results are shown in Table 4. We see that our method is significantly superior to the Sub-Method 1 and 2 because the p-value is less than 0.05. But for Sub-Method 3, the H value is 0. The reason can be explained as follows: from Section IV-D, we can see that the ACC of the first model trained by Sub-Method 3 exceeds 95%, which is an outlier, better than our ensemble method. However, after using the mean value of multiple results, our method is more stable and generalization is better.

To evaluate the ablation process more intuitively, we obtain the ROC curves of the three Sub-Methods in Fig. 7. It can be seen that, as the ablation process being proceeded, the AUCs of the three Sub-Methods show an obviously increasing trend.

And Fig. 8 shows the Grad-CAM [39] visualization results of the three Sub-Methods. Since our method is an ensemble learning process, only the visualization results of the Sub-Methods are shown. In order to

Table 5
The summary of differences between various methods.

Method	Summary
Self-Trans [16]	Implemented on the Ori-COVID-CT dataset. Relatively complex pretraining methods are designed, and high recognition accuracy (0.86) is achieved.
Wang et al. [8]	Implemented on the Ori-COVID-CT and another public chest CT dataset. By redesigning the previous network architecture, and leveraging multi-task learning and joint learning, two datasets are trained simultaneously. Although its recognition accuracy on the Ori-COVID-CT dataset is not high, it breaks through the data heterogeneity.
Evidential Covid-Net [9]	Implemented on the Ori-COVID-CT dataset. Deep features and belief function are combined. It uses 50% of the training labels, i.e., it is trained by semi-supervised learning. And it shows better performance than the baseline ResNet50 model.
Mishra et al. [13]	Implemented on the Ori-COVID-CT dataset. It is a simple but efficient method. The classification results come from the decision fusion of multiple individual convolutional neural networks. Among several competitive methods, it achieves the highest recognition accuracy (0.8834).
Ewen et al. [17]	Implemented on the Ori-COVID-CT dataset. It uses a simple self-supervised learning strategy. Even if a few data are used for pretraining, promising performance can be achieved.
LA-DNN [12]	Implemented on the Exp-COVID-CT and Exp-COVID-CT* datasets. Even if there is no more complex method design, only the multi-label binary classification task is performed, excellent results are achieved. And the network converges very fast. This results in a much shorter training time.
Our method	Implemented on the Exp-COVID-CT dataset. Within our method, the characteristics of the data itself are mined, loss function is carefully designed, pretraining method is constructed, and an ensemble learning technique is utilized. High recognition accuracy on small-scale dataset has been achieved with our method.

better reflect the performance differences between different algorithms, we select a few images with manual annotation from the dataset, as can be seen from the first three CT images in column (1). Through visualization, the three Sub-Methods have achieved good performance in the first three CT images. The visualization of the last three images shows that: 1) Sub-Method 1 without the LSFLLW-R and the ensemble learning techniques could not accurately locate the lesion area; 2) Sub-Method 3 can better capture the lesion region, and the area of interest is smaller and more accurate, compared to that captured by method Sub-Method 2. They all validate the effectiveness of our design at each step.

4.4. LSFLLW-R for model stability

In nine repeated experiments of the same method, we also observe two interesting phenomena. First, we consider Sub-Method 2, i.e., we directly apply the pretrained model to the multi-label Exp-COVID-CT dataset for supervised fine-tuning without regularization. The results produce instability. The metric of one model is much lower than that of the other 8 models. Take ACC, F1, and PRE, for instance, we draw Fig. 9. (a). The lower is the corresponding box plot of the ACC. As can be seen, the three metrics of the fourth model are very poor. Compared with other models, there are great fluctuations. And the box plot clearly shows that the Sub-Method 2 yields the outlier of ACC. However, the model trained with LSFLLW-R is very stable. Fig. 9. (b) shows the same three metrics of the nine models generated by Sub-Method 3 and the corresponding box plot of the ACC. The difference between Sub-Method 3 and Sub-Method 2 is the introduction of LSFLLW-R. By comparing Fig. 9. (a) and 9. (b), we can obviously find that the addition of LSFLLW-R makes the training or model more stable and robust. These results further validate that LSFLLW-R not only improves the diagnostic performance of the model, but also makes the training smoother. Second, it is worth noting that the first model of Sub-Method 3 achieves an ACC of 0.966, which is much higher than that of our proposed method. However, in terms of the average performance of the Sub-Method 3 and our proposed method, a single model is not representative. That is to say, the overall generalizability of the Sub-Method 3 is not as good as our proposed method.

Based on the analysis in Sections B, C, and D above, first, the highest mean classification performance is achieved by our proposed method. Second, considering all metrics, our proposed method achieves the most balanced results. Third, the model we obtained is the most stable.

5. Conclusions and future works

When faced with challenge of computer-aided recognition of COVID-19, the existing methods usually cannot achieve adequately high recognition accuracy, while require a lot of training data, simultaneously. Aiming at addressing these two issues, in this work, we build a high-performance COVID-19 recognition system on basis of the small-scale multi-label Exp-COVID-CT dataset. First, we leverage the two-phase CSSL-based pretraining to obtain a base model with good representation learning ability. Then, according to the characteristics of this dataset, we reasonably and skillfully design multiple regularizations to continuously optimize the solution space during the supervised training phase. Finally, through the specific ensemble learning technique, the generalizability, balance, and stability of the whole system are further improved. Our proposed system achieves promising COVID-19 recognition results on this small dataset, with the values of the accuracy, F1-score, and the AUC reaching 94.3%, 94.7%, and 98.9% respectively. Experimental results demonstrate that the developed system can locate the disease area precisely, exhibiting superior detection performance to several other several state-of-the-art COVID-19 recognition approaches.

However, after careful and in-depth analysis, we find out that there exist two limitations in the proposed COVID-19 recognition system, required to be further ameliorated. The first limitation is that, from the initial model to the generation of the final system, our developed system needs relatively long training time, especially the two-phase CSSL procedure. Besides, our presented system is established on the strength of a single dataset, which might ignore the heterogeneity of data caused by different imaging conditions, and might result in relatively imperfect classification performance on heterogeneous CT images. Ulteriorly, we compare the differences between our designed system and several other competitive state-of-the-art methods, with the differences being summarized in Table 5.

As for the future work, directing at the above analyzed defects of our proposed system, data heterogeneity will be studied and considered in our future work. Additionally, we will focus on the research of more rational and effective strategy design for the self-supervised learning paradigm, so as to reduce the training time and mine more valuable information from the data itself in the case of a more limited amount of data.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 61473150.

References

[1] T. Ozturk, M. Talo, E.A. Yildirim, U.B. Baloglu, O. Yildirim, U.R. Acharya, Automated detection of COVID-19 cases using deep neural networks with X-ray images, Comput. Biol. Med. 121 (2020), 103792.

[2] M.E.H. Chowdhury, et al., Can AI help in screening viral and COVID-19 pneumonia? IEEE Access 8 (2020) 132665–132676.

[3] L. Wang, A. Wong, COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images, 2020 *arXiv: 2003.09871*, [Online]. Available: https://arxiv.org/abs/2003.09871.
[4] Y. Oh, S. Park, J.C. Ye, Deep Learning COVID-19 Features on CXR Using Limited Training Data Sets, 2020 arXiv:2004.05758, [Online]. Available: https://arxiv.org/abs/2004.05758.
[5] J. Zhang, et al., Viral pneumonia screening on chest X-ray images using confidence-aware anomaly detection, arXiv e-prints:2003.12338, https://arxiv.org/abs/2003.12338, 2020 [Online]. Available:.
[6] A. Bernheim, X. Mei, M. Huang, Y. Yang, M. Chung, Chest CT findings in Coronavirus disease-19 (COVID-19): relationship to duration of infection, Radiology 295 (3) (2020), 200463.
[7] L. Sun, et al., Adaptive feature selection guided deep forest for COVID-19 classification with chest CT, IEEE.J.Biomed.Health Inf. 24 (2020) 2798–2805.
[8] Z. Wang, Q. Liu, Q. Dou, Contrastive cross-site learning with redesigned net for COVID-19 CT classification, IEEE.J.Biomed.Health Inf. 24 (2020) 2806–2813.
[9] L. Huang, S. Ruan, T. Denoeux, Covid-19 Classification with Deep Neural Network and Belief Functions, 2021 arXiv:2101.06958, [Online]. Available: https://arxiv.org/abs/2101.06958.
[10] T. Javaheri, M. Homayounfar, Z. Amoozgar, R. Reiazi, R. Rawassizadeh, CovidCTNet: an Open-Source Deep Learning Approach to Identify Covid-19 Using CT Image, 2020 arXiv:2005.03059, [Online]. Available: https://arxiv.org/abs/2005.03059.
[11] X. Gao, Y. Qian, A. Gao, COVID-VIT: Classification of COVID-19 from CT Chest Images Based on Vision Transformer Models, 2021 arXiv: 2107.01682, [Online]. Available: https://arxiv.org/abs/2107.01682.
[12] B. Liu, X. Gao, M. He, L. Liu, G. Yin, A fast online COVID-19 diagnostic system with chest CT scans, in: " in Proceedings of ACM Knowledge Discovery and Data Mining vol. 2020, SIGKDD), San Diego, CA, USA, 2020.
[13] A.K. Mishra, S.K. Das, P. Roy, S. Bandyopadhyay, Identifying COVID19 from chest CT images: a deep convolutional neural networks based approach, J.Healthc.Eng. 2020 (2020).
[14] A. Amyar, R. Modzelewski, H. Li, S. Ruan, Multi-task deep learning based CT imaging analysis for COVID-19 pneumonia: classification and segmentation, Comput. Biol. Med. 126 (2020), 104037.
[15] O. Gozes, et al., Rapid AI Development Cycle for the Coronavirus (COVID-19) Pandemic: Initial Results for Automated Detection & Patient Monitoring Using Deep Learning CT Image Analysis, 2020 arXiv e-prints:2003.05037, [Online]. Available: https://arxiv.org/abs/2003.05037.
[16] X. He, X. Yang, S. Zhang, J. Zhao, P. Xie, Sample-efficient deep learning for COVID-19 diagnosis based on CT scans, medRxiv (2020), 04.13.20063941, 2020. [Online]. Available: https://www.medrxiv.org/content/10.1101/2020.04.13.20063941v1.
[17] N. Ewen, N. Khan, Targeted self supervision for classification on a small COVID-19 CT scan dataset, in: IEEE International Symposium on Biomedical Imaging (ISBI), 2021, pp. 1481–1485. Nice, France.
[18] R. Hadsell, S. Chopra, Y. Lecun, Dimensionality reduction by learning an invariant mapping, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR), New York, NY, USA, 2006.
[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 2818–2826.
[20] T.Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, IEEE Trans. Pattern Anal. Mach. Intell. 42 (2017) 318–327.
[21] F. Shi, et al., Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for COVID-19, IEEE .Rev. Biomed. Eng. 14 (2020) 4–15.
[22] X. Yang, X. He, Y. Liang, Y. Yang, S. Zhang, P. Xie, Transfer Learning or Self-Supervised Learning? A Tale of Two Pretraining Paradigms, 2020 *arXiv: 2007.04234*, [Online]. Available: https://arxiv.org/abs/2007.04234.
[23] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, R. Shah, Signature verification using a "Siamese" time delay neural network, in: Conference on Neural Information Processing Systems, NeurIPS), Denver, Colorado, USA, 1993, pp. 737–744.
[24] X. Chen, H. Fan, R. Girshick, K. He, Improved Baselines with Momentum Contrastive Learning, 2020 *arXiv:2003.04297*, [Online]. Available: https://arxiv.org/abs/2003.04297.
[25] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020. Seattle, WA, USA.
[26] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning (ICML), Virtual Event, 2020, pp. 1597–1607.
[27] J.-B. Grill, et al., Bootstrap your own latent-a new approach to self-supervised learning, in: Conference on Neural Information Processing Systems (NeurIPS), 2020. Online Event.
[28] M. Caron, et al., Emerging properties in self-supervised vision transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Canada, Montreal, BC, 2021, pp. 9650–9660.
[29] X. Chen, S. Xie, K. He, An empirical study of training self-supervised vision transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Canada, Montreal, BC, 2021, pp. 9640–9649.
[30] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, N. Houlsby, An image is worth 16x16 words: transformers for image recognition at scale, in: International Conference on Learning Representations (ICLR), Virtual Event, Austria, 2021.
[31] A. Van den Oord, Y. Li, O. Vinyals, Representation Learning with Contrastive Predictive Coding, 2018 *arXiv e-prints:1807.03748*, https://arxiv.org/abs/1807.03748.
[32] G. Huang, Z. Liu, L.V.D. Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2261–2269. Honolulu, HI, USA.
[33] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, F.F. Dept, ImageNet : a large-scale hierarchical image database, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR), Miami, Florida, USA, 2009.
[34] T.G. Dietterich, Ensemble methods in machine learning, in: International Workshop on Multiple Classifier Systems (MCS), Cagliari, Italy, 2000, pp. 1–15.
[35] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[36] R.E. Schapire, The strength of weak learnability, Mach. Learn. 5 (2) (1990) 197–227.
[37] D.H. Wolpert, Stacked generalization, Neural Network. 5 (2) (1992) 241–259.
[38] P.N. Tan, M. Steinback, V. Kumar, Introduction to Data Mining, Pearson Education, Inc., 2006.
[39] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 618–626. Venice, Italy.