Pattern Recognition 113 (2021) 107826

Contents lists available at ScienceDirect

Pattern Recognition

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

Momentum contrastive learning for few-shot COVID-19 diagnosis from chest CT images

Xiaocong Chen[a,*], Lina Yao[a], Tao Zhou[b], Jinming Dong[a], Yu Zhang[c]

[a] School of Computer Science and Engineering at University of New South Wales, NSW 2052, Australia
[b] Inception Institute of Artificial Intelligence, Abu Dhabi, UAE
[c] Department of Bioengineering, Lehigh University, Bethlehem, PA 18015, USA

ARTICLE INFO

Article history:
Received 19 June 2020
Revised 13 November 2020
Accepted 22 November 2020
Available online 16 January 2021

Keywords:
COVID-19 diagnosis
Few-shot learning
Contrastive learning
Chest CT images

ABSTRACT

The current pandemic, caused by the outbreak of a novel coronavirus (COVID-19) in December 2019, has led to a global emergency that has significantly impacted economies, healthcare systems and personal wellbeing all around the world. Controlling the rapidly evolving disease requires highly sensitive and specific diagnostics. While RT-PCR is the most commonly used, it can take up to eight hours, and requires significant effort from healthcare professionals. As such, there is a critical need for a quick and automatic diagnostic system. Diagnosis from chest CT images is a promising direction. However, current studies are limited by the lack of sufficient training samples, as acquiring annotated CT images is time-consuming. To this end, we propose a new deep learning algorithm for the automated diagnosis of COVID-19, which only requires a few samples for training. Specifically, we use contrastive learning to train an encoder which can capture expressive feature representations on large and publicly available lung datasets and adopt the prototypical network for classification. We validate the efficacy of the proposed model in comparison with other competing methods on two publicly available and annotated COVID-19 CT datasets. Our results demonstrate the superior performance of our model for the accurate diagnosis of COVID-19 based on chest CT images.

1. Introduction

The latest coronavirus, COVID-19, was initially reported in Wuhan, China toward the end of 2019 and has since spread rapidly around the globe, leading to a worldwide crisis. As an infectious lung disease, COVID-19 leads to severe acute respiratory distress syndrome (ARDS) and is accompanied by a series of side effects that include a dry cough, fever, tiredness, shortness of breath, etc. As of October 18th 2020, more than 39 million individuals around the world have been confirmed as having COVID-19, with a roughly 6.3% case fatality rate, according to the World Health Organization.[1]

So far, no effective treatment for COVID-19 has been found. One of the major hurdles is the lack of efficient diagnostic methods. Therefore, an accurate and rapid diagnosis platform is urgently required to conduct COVID-19 screening and prevent its further spread. Currently, most tests are based on real-time reverse transcriptase polymerase chain reaction (RT-PCR). However, each RT-PCR test can take several hours to produce results. With the current spread rate of COVID-19, this is not acceptable. Further, the limited number of test kits exacerbates the situation [1–3]. Recent studies also show that the RT-PCR suffers from low sensitivity and accuracy, often requiring repeated entries [4,5]. This prevents patients from being confirmed in a timely manner, increasing the potential risk of spreading.

In order to address these challenges, scientists around the world are trying to develop new diagnostic systems. Some studies [6,7] have demonstrated that chest computed tomography (CT) imaging can help in diagnosing COVID-19 rapidly. Salehi et al. [8] concluded that chest CT imaging is sensitive for diagnosing COVID-19 even when patients do not have clinical symptoms. Specifically, three typical radiographic features, including consolidation, pleural effusion and ground class opacification, can be easily observed from the CT images of COVID-19 patients [9,10].

With this in mind, several methods based on chest CT images been developed for diagnosing COVID-19. For instance, some studies used a 3D CNN to diagnosis of COVID-19 from chest CT scans [11]. Mei et al. [12] adopted ResNet to rapidly iden-

* Corresponding author.
E-mail addresses: xiaocong.chen@unsw.edu.au (X. Chen), lina.yao@unsw.edu.au (L. Yao), taozhou.ai@gmail.com (T. Zhou), yuzi20@lehigh.edu (Y. Zhang).

[1] https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports

https://doi.org/10.1016/j.patcog.2021.107826
0031-3203/© 2021 Elsevier Ltd. All rights reserved.

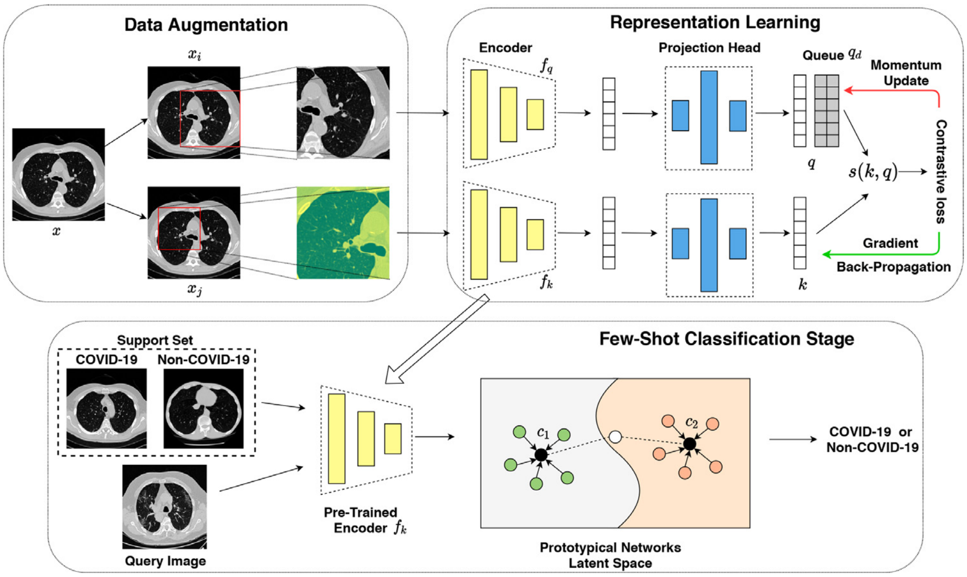

Fig. 1. The overall architecture of our approach. Top: The pre-training stage, which includes data augmentation and representation learning. The pretext task is an instance discrimination task. Bottom: Few-shot classification with 2-way, 1-shot example. For classification, the support images and query image are encoded by the pre-trained encoding network. Query sample embeddings are compared with the centroid of training sample embeddings and used to fine-tune the pre-trained encoder.

tify COVID-19. Besides diagnosis, several works also used the segmentation techniques for detection [13,14]. However, all existing methods are trained using the limited samples available from a small number of patients and may not generalize well to new patients. It is well-known that a lack of labelled training data is a common challenge since deep learning methods generally require a large volume of data for accurate training. Significant research efforts have been dedicated to alleviating this problem through, for example, data augmentation or generative adversarial networks (GANs) [15–18]. However, these methods are highly sensitive to parameter selection. Hand-tuned data augmentation methods like rotation may lead to overfitting [19], images generated by a GAN cannot simulate the real patient data which, may introduce unpredictable bias in the testing phase [15]. Recently, few-shot learning attracted significant attention in medical image analysis. In general, few-shot learning aims to leverage existing data to classify new tasks from similar domains. The basic workflow for few-shot learning is to pre-train an embedding network on a large dataset (e.g. ImageNet), then fine-tune the weights of this network, and finally apply it to a small unseen dataset [20,21]. However, the performance is only marginally improved this way. One reason lies in that ImageNet contains a broad range of categories and pre-training on this dataset often introduces irrelevant information, which does not help in learning effective embeddings for improving lung-specific feature representation. In addition, pre-training on ImageNet incurs a high computational cost; for example, ImageNet-1B typically required over 50 GPU days.

To address this challenge, we develop an end-to-end trainable deep few-shot learning framework that can provide accurate predictions with minimal training on chest CT images. Specifically, we first use the instance discrimination task to enforce model to discriminate two images are the same instance or not. Different views of the same images are then generated to augment the original dataset. As our goal at this stage is to increase variances other than discrimination, we are able to effectively avoid the disadvantages of data augmentation mentioned previously.

We then deploy a self-supervised strategy [22] powered with momentum contrastive training to further boost the performance. The key idea is to build a dynamic dictionary to perform (key, query) look-up, where the keys are sampled from data and encoded by the encoder. However, the key in the dictionary is noisy and inconsistent due to the back-propagation [23]. We apply the momentum mechanism to mitigate this effect by updating the key and query encoders at different scales. Finally, we utilize two public lung datasets to pre-train an embedding network and employ the prototypical network [24] to conduct the few-shot classification, which learns a metric space for classification by measuring the distances to the derived prototypical representation of each class. Extensive experiments on two new datasets demonstrate that our model provides a promising tool for quick COVID-19 diagnosis with very limited available training data.

2. Problem Definition

Due to the shortage of annotated COVID-19 CT images, normal classification methods may not work properly. As such, we formulate COVID-19 diagnosis as a few-shot classification problem. Few-shot learning is designed for cases in which only a few samples of new class are available for classifier training. It can be defined as a M-way, C-shot episodic task [25], where M represents the number of classes and C represents number of samples available for each class. The training set, which has never been seen before, can be represented as $D = \{(x_0, y_0), \ldots, (x_d, y_M)\}$, where d is the number of samples in the dataset. We randomly select the support set and query set from D: (i) The support set S can be partially or fully made up of M classes but only contain $C+1$ samples. (ii) We randomly select one sample from the $C+1$ samples to form the test set (query set). Hence, COVID-19 diagnosis can be represented as a two-way, C-shot learning problem.

3. Methodology

In this section, we will introduce our proposed self-supervised COVID-19 diagnosis method. The overall flowchart is illustrated in Fig. 1. We will describe the three major components of our method, which include data augmentation, representation learning and the few-shot classification.

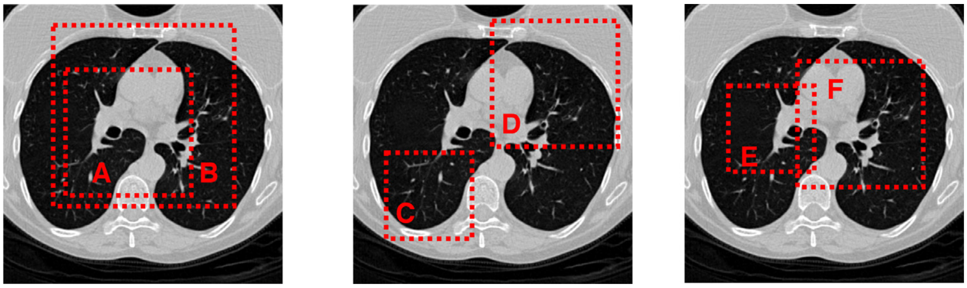

Fig. 2. Three possibilities for random cropping. Dashed boxes are augmented views. Crops A, C, E will have random color distortions applied, while B, D, F will not change if method (2) is chosen. All the cropped sections will be resized back to the original input image size. For the instance discrimination task, the goal, given B, D, F is to determine whether or not A, C, E are in the same instance.

3.1. Data Augmentation

Data augmentation has been widely used in unsupervised representation learning and supervised learning [26,27]. A few existing approaches define the contrastive classification task as changing the structure of images. For instance, Hjelm et al. [28] and Bachman et al. [29] used global-to-local view for contrastive learning as shown in the first example in Fig 2. Meanwhile, Oord et al. [22] and Henaff et al. [30] achieved neighbor prediction using the adjacency view (middle example, Fig 2). An overlapping view of the two approaches can be seen in third image of Fig 2.

In this study, we apply a stochastic data augmentation $\mathcal{T}$ which randomly transfers a given example image x into two different views, denoted as x_i, x_j. We consider the pair x_i, x_j as positive. Further, we apply two simple augmentation strategies in sequence: (1) random cropping, followed by a resizing operation back to the original size with random flipping; or (2) random cropping with color distortions followed by a resizing operation. When a new image is fed into the model, one of the above methods is randomly selected for augmentation. This process is repeated twice to generate two different views. Note that we implement color distortions using the torchvision[2] package in PyTorch [31].

3.2. Contrastive Visual Embedding

Using contrastive learning to learn visual embeddings was first explored by Hadsell et al. [32]. Given an image set $\{\mathcal{I} = i_1, \ldots, i_p\}, x_i \in \mathbb{R}^d$, the goal of the task is to find a mapping function $G : \mathbb{R}^d \mapsto \mathbb{R}^a, a \ll d$ that satisfies:

$$s(G(x), G(x^+)) \gg s(G(x), G(x^-)) \tag{1}$$

where $s(\cdot,\cdot)$ is a function used to measure the similarity between two inputs. G is designed for dimension reduction and representation learning. Finally, x^+, x^- represent the positive and negative samples, where x^+ is similar to x and x^- is dissimilar. It is worth mentioning that the contrastive learning is a type of unsupervised learning. A simple framework for contrastive learning was proposed by Chen et al. [33]. Specifically, the representations are learned by maximizing the agreement between differently augmented views x_i, x_j of the same data example x via a contrastive loss in the latent space. We adopt this framework in our model. Specifically, our representation learning stage consists of three modules: the encoder, projection head and, contrastive loss function.

Encoder The neural network based encoder $f(\cdot)$ can extract representations from the augmented images. This framework is flexible for adopting any type of network architecture without constraints. In this study, we adopt ResNet [34] to obtain the representation h_i, h_j, $h_i = f(x_i) = \mathrm{ResNet}(x_i)$ where h_i is the $\mathbb{R}^d$ output of the average pooling layer.

Projection head The projection head $g(\cdot)$ is a function that can map the resulting representation into the application space of the contrastive loss. The most common projection head used is the multilayer perceptron (MLP) with one hidden layer [33]. In this case, we can express the z_i (as well as z_j) as:

$$z_i = g(h_i) = W^2\sigma(W^1 h_i) \tag{2}$$

where W^1, W^2 are the weights of the hidden layer and output layer, respectively. The $\sigma(\cdot)$ is the non-linear ReLU activation function, which can be defined as:

$$\mathrm{ReLU}(x) = \begin{cases} 0 & x \le 0 \\ x & x > 0 \end{cases} \tag{3}$$

We will examine the effectiveness of this projection head in Section 4.

Contrastive loss function The contrastive loss function is defined for the contrastive pre-text task. It was first proposed by Hadsell et al. [32] and is used to calculate the value when the query is similar to the positive key and dissimilar for all other keys. In this manuscript, we only consider the instance discrimination task [35]. Given a set $\{x_k\}$ including a positive pair x_i, x_j, the contrastive task aims to identify x_j in set $\{x_k\}_{k\neq i}$ for a given x_i.

We define the contrastive task on pairs of augmented images from a randomly selected minibatch with N samples. The augmentation process results in $2N$ data points. To create the contrastive task, we need enough negative samples to construct the loss function. Similar to Doersch et al. [36], we treat the other $2N-2$ examples as the negative samples. The similarity function $s(\cdot,\cdot)$ can be defined as the cosine similarity:

$$s(v, u) = \frac{v^\intercal u}{\|v\|\|u\|} \tag{4}$$

where v, u are two vectors. Based on this, we can define the loss function for a pair of positive samples (i, j) as:

$$\mathcal{L}_{i,j} = -\log \frac{\exp(s(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{k\neq i} \exp(s(z_i, z_k)/\tau)} \tag{5}$$

Here $\mathbb{1}_{k\neq i} \in \{0, 1\}$ is the indicator which has a value of 1 when $k \neq i$ and 0 otherwise, and τ is the temperature parameter. This loss is known as the normalized temperature-scaled cross entropy loss [22,29]. However, Eq. (5) only considers the positive samples and ignores negative samples. Note that, the margin based contrastive loss function [32] has the same problem only considering about the positive keys. This may lead to potential bias. To avoid this, we introduce the momentum mechanism into our model.

Contrastive learning can also be expressed as training an encoder to conduct a dictionary lookup task. Consider an encoded query q and encoded samples $x_i, \ldots, x_k$, which are the keys of the

[2] https://pytorch.org/docs/stable/torchvision/index.html

dictionary. If the query q is similar to the sample x^+, there is a match. For the negative samples x^-, there is no match in the dictionary. Based on this definition, He et al. proposed an unsupervised learning-based framework MoCo [23], by adopting contrastive learning.

Based on the above definition, the goal of contrastive learning is to build a discrete dictionary for high-dimensional continuous inputs. The core of MoCo is to maintain a dictionary with a queue. The benefit of this is that the encoder can reuse the encoded keys from the previous mini-batch. In addition, the dictionary can be much larger than the mini-batch and easy to adjust. As the number of samples that can be included in the dictionary is fixed, once the dictionary is full, it will progressively remove the oldest records. In this way, the consistency of the dictionary can be maintained as the oldest samples are often out-of-date and inconsistent with the new entries. Another approach, called Memory Bank [35], tries to store the historical records of the encoded samples. This approach maintains a bank of all the representations of the dataset. The dictionary then randomly samples from the memory bank directly for each mini-batch without back-propagation. However, this method will lead to inconsistency when sampling. To overcome this, back-propagation should be conducted to keep the sampling step up-to-date. A simple solution is to copy the key encoder f_k from the query encoder f_q without the gradient. However, the encoder changes constantly, which can lead to a noisy key representation and poor results. The momentum contrast was introduced to address this problem, using a different method to update the gradient for f_k:

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q. \tag{6}$$

where θ_k is the parameter for f_k, θ_q is the parameter for f_q and $m \in [0, 1)$ is the momentum coefficient. We use back-propagation to update the parameter θ_q and use Eq. (6) to update θ_k. Benefiting from the momentum coefficient, the update of θ_k is smoother than θ_q. According to the different update strategies, the query and key will eventually be encoded by different encoders.

Based on the above discussion, we use the dictionary as a queue to allow the encoder to reuse the previous encoded sample. The loss function for the pre-trained model can be written as:

$$\mathcal{L} = -\log \frac{\exp(q, k^+)/\tau}{\exp(q, k^+)/\tau + \sum_{k^-} \exp(q, k^-)/\tau}. \tag{7}$$

Different from Eq. (5), here we need to consider the queue and the negative cases, so we slightly modify the loss function to fulfil this requirement by introducing the positive examples k^+ and negative examples k^-, where $q_k = k^+ \cup k^-$. In the instance discrimination pre-text task, a positive pair is formed when a query q and a key k are augmented from the same sample; otherwise, a negative pair is created. Once the pre-training step is finished, we extract the pre-trained encoder $f(\cdot)$ and integrate it into our classification module. It is worth mentioning that the triple loss [37] is another popular loss function that considers both positive and negative examples. However, the triple loss does not converge easily and is time-consuming. Hence, it is normally only used in identification [38,39] or fine-grained image classification tasks. For our task, using the contrastive loss is adequate as we are dealing with instance discrimination task.

3.3. Prototypical network for few-shot classification

Another step in our workflow is classification. In this stage, meta-learning is applied to fine-tune the pre-trained encoder to fit the class changes required by few-shot learning. Then we use Prototypical Networks [24] for few-shot classification. The prototypical network learns an embedding that maps all inputs into a mean vector c in the latent space to represent each class. The goal of the pre-trained encoder is to ensure that similar images are close and dissimilar images are separate in the latent space. The prototypical network has a similar goal, so it is used to fine-tune our pre-trained encoder. For class m, the centroid embedding features can be written as:

$$c_m = \frac{1}{|S|} \sum_{(x_d, y_M) \in S} \psi(x_d) \tag{8}$$

where $\psi(\cdot)$ is the embedding function from the prototypical network. As the prototypical network is a metric based learning method, we use the Euclidean distance to produce the distribution for all classes for a query q.

$$p(y = m|q) = \frac{\exp(-d(\psi(q), c_m))}{\sum_{m'} \exp(-d(\psi(q), c_{m'})}. \tag{9}$$

Eq. (9) is based on the softmax function over the distance between a query set's embedding and the features of the class. The loss function for this stage can be defined as:

$$\mathcal{L}_{\text{meta}} = d(\psi(q), c_m) + \log(d(\psi(q), c_{m'})) \tag{10}$$

3.4. Training strategy

Algorithm 1 shows the whole pre-training workflow of our model.

Algorithm 1: Training algorithm for the pre-training.

```
input : Batch size $N, \tau, f_k, f_q, g, \mathcal{T}, q_k$
for sampled mini-batch $\{x_k\}_{k=1}^{N}$ do
    for $k \in \{1, \dots, N\}$ do
        Select two data augmentation functions from $\mathcal{T}$: $t, t'$ ;
        $x_{2k-1} = t(x_k), \hat{x}_{2k-1} = t'(x_k)$ ;
        $h_{2k-1} = f_k(x_{2k-1}), h_{2k} = f_q(\hat{x}_{2k-1})$ ;
        $z_{2k-1} = g(h_{2k-1}), z_{2k} = g(h_{2k})$ ;
    end
    for $i \in \{1, \dots, 2N\}, j \in \{1, \dots, 2N\}$ do
        Calculate the similarity using Eq. (4) ;
    end
    Update $f_k$ to minimize Eq. (7);
    Update $f_q$ by Eq. (6) ;
    enqueue($q_k$, $z_{2k-1}$) ;
    dequeue($q_k$);
end
return $f_k$;
```

4. Experiments

4.1. Datasets

We evaluated our proposed model using two publicly available annotated COVID-19 CT slices datasets: (1) COVID-19 CT[3] and (2) a dataset provided by the Italian Society of Medical and Interventional Radiology[4] and preprocessed by MedSeg.[5] It is worth mentioning that there is no overlap between COVID-19 CT and MegSeg as they come from different countries. When dividing the support and query sets for classification, we divided the datasets at a patient-level instead of CT level to avoid any possible overlap. The basic statistics for the COVID-19 CT dataset and MegSeg are

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

[4] https://www.sirm.org/category/senza-categoria/covid-19/

[5] http://medicalsegmentation.com/covid19/

Table 1
Number of patients and number of CT slices available in the experimental datasets.

		COVID-19 CT	MegSeg
# of Patients	COVID-19	216	43
	Non-COVID-19	171	0
# of CT Slices	COVID-19	349	110
	Non-COVID-19	397	0

summarized in Table 1. We combined the two datasets for testing. Note that all CT slices were resized to 512×512 using opencv2.[6]

A proper pre-training is required for our proposed model. Different from other existing methods, such as Self-Trans [20], that used ImageNet to pre-train the model, we adopted DeepLesion [40] and the Lung Image Database Consortium Image Collection (LIDC-IDRI).[7] DeepLesion contains over 32,000 lung CT images while LIDC-IDRI includes 244,617 ones. Both datasets are public and focus on lung diseases. We used the two datasets without labels to pre-train the encoder network.

4.2. Experimental settings

For pre-training, we used the SGD optimizer with a weight decay of 0.0001 and momentum of 0.9. The momentum update coefficient was 0.999. The mini-batch size was set to 256 in eight GPUs. The number of epochs was 200. The initial learning rate was 0.03, which was then multiplied by 0.1 after 120 and 160 epochs, as described in [35]. ResNet-50 was used as the encoder. The two-layer MLP projection head included a 2048-D hidden layer with a ReLU activation function. The weights were initialized by using He initialization [41], and the temperature parameter τ was set to 0.07. For the classification stage, we followed the default settings of the prototypical net. The experiments were conducted on eight GPUS which includes six NVIDIA TITAN X Pascal GPUs and two NVIDIA TITAN RTX.

4.3. Evaluation and results

We evaluated model performance using four metrics: (i) Accuracy, which measures the percentage of correctly classified samples over the whole dataset; (ii) Precision, which measures the percentage of true positives (TP) over all predicted positive samples; (iii) Recall, used to measure the percentage of TPs over all positive samples; and (iv) Area-under-the-curve (AUC) which measures the relation between FPs and TPs. We trained and tested each of the compared methods on COVID-19 CT and MegSeg dataset using 10-fold cross-validation at a patient-level with the cross-entropy loss function.

The experimental results are summarized in Table 2. We found that the designed two-way, one-shot model yielded very similar performance to ResNet-50. In addition, we found that the obtained classification performance is worse when the model is pre-trained on ImageNet. As discussed previously, an extra step may be required to conduct transfer learning from common items to lung CT slices.

As previously mentioned, our method used few-shot learning. We were thus interested to see how varying the number of shots would affect the model performance. Accordingly, we conducted an experimental analysis to explore the relationship between the classification performance and the number of shots. The results are shown in Table 3, where ResNet-50 is used as a baseline method for the comparison. As can be seen, the classification performance of our model is gradually improved with the increase in the number of shots. Specifically, our model achieved significantly improved performance when using four shots compared with one shot and outperformed ResNet-50, but no obvious further improvement was observed when using more than five shots. These results indicate that the pre-trained encoder effectively captured the features from unknown images to improve the classification performance. Additionally, we provide visualizations of the features learned by different methods including our method, Pretrain with ImageNet, DenseNet-121 and ResNet-50 in Fig 3. Here, both DenseNet-121 and ResNet-50 were directly trained on the COVID-19 dataset described in Table 1. As can be seen, our method learned more features that focused on the lung area, improving the classification accuracy in comparison with approaches. Table 4 summarizes the training time taken by different methods for a comparison of the computational cost. As our method was not trained on the COVID-19 dataset, the corresponding training time was not available. The method trained on ImageNet took about 150 h.

4.4. Ablation study

In this section, we conducted extensive ablation studies to demonstrate the importance of each component in our model. The default setting of our method was two-way, one-shot and ResNet-50 used the same setting as in the previous section. We investigated the following research questions: (1) How would data augmentation and projection head affect the performance? (2) How important is the fine-tuning stage? (3) How would the resizing operation affect the performance? (4) Is the result significantly affected when using a different encoder?

First, we explored the role of the data augmentation and projection head. We conducted the experiments on our model without augmentation and without projection head, respectively.

The results summarized in Table 5 show that data augmentation had a significant effect on the model performance while the projection head yielded only a slight improvement. One possible reason why the projection head was not able to provide an obvious improvement would be that it was only used to extract the most important information from the similarity vector. As such, it simply worked as a filter without introducing additionally useful features.

In addition, we also investigated that the impact of different data augmentation strategies. Specifically, we compared the classification results between random-cropping of using three augmentation strategies and only using a single one. All the com-

Table 2
Performance comparison between our proposed model and other methods. Our model and the method pretrained on ImageNet use a two-way, one-shot strategy.

	Accuracy	Precision	Recall	AUC
ResNet-50	0.873 ± 0.013	0.894 ± 0.012	0.874 ± 0.012	0.935 ± 0.014
DenseNet-121	0.855 ± 0.013	0.867 ± 0.012	0.859 ± 0.012	0.894 ± 0.013
ImageNet Pre	0.732 ± 0.023	0.744 ± 0.021	0.738 ± 0.023	0.870 ± 0.022
Ours	0.868 ± 0.012	0.883 ± 0.011	0.872 ± 0.012	0.931 ± 0.013

[6] https://opencv.org/

[7] https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI

Table 3
Classification performance when using different settings in our model. Here, *W* indicates the number of ways and *S* the number of shots. For instance, 2W1S represents the two way, one shot setting. ResNet-50 is used as a baseline method for comparison.

	Accuracy	Precision	Recall	AUC
ResNet-50	0.873 ± 0.013	0.894 ± 0.012	0.874 ± 0.012	0.935 ± 0.014
Ours(2W,1S)	0.868 ± 0.012	0.883 ± 0.011	0.872 ± 0.012	0.931 ± 0.013
Ours(2W,2S)	0.872 ± 0.012	0.890 ± 0.011	0.875 ± 0.012	0.935 ± 0.012
Ours(2W,3S)	**0.876 ± 0.012**	**0.895 ± 0.011**	**0.878 ± 0.011**	**0.938 ± 0.012**
Ours(2W,4S)	**0.881 ± 0.012**	**0.898 ± 0.011**	**0.882 ± 0.011**	**0.942 ± 0.012**
Ours(2W,5S)	**0.884 ± 0.010**	**0.899 ± 0.010**	**0.885 ± 0.012**	**0.946 ± 0.010**
Ours(2W,6S)	**0.885 ± 0.011**	**0.899 ± 0.012**	**0.886 ± 0.010**	**0.945 ± 0.009**

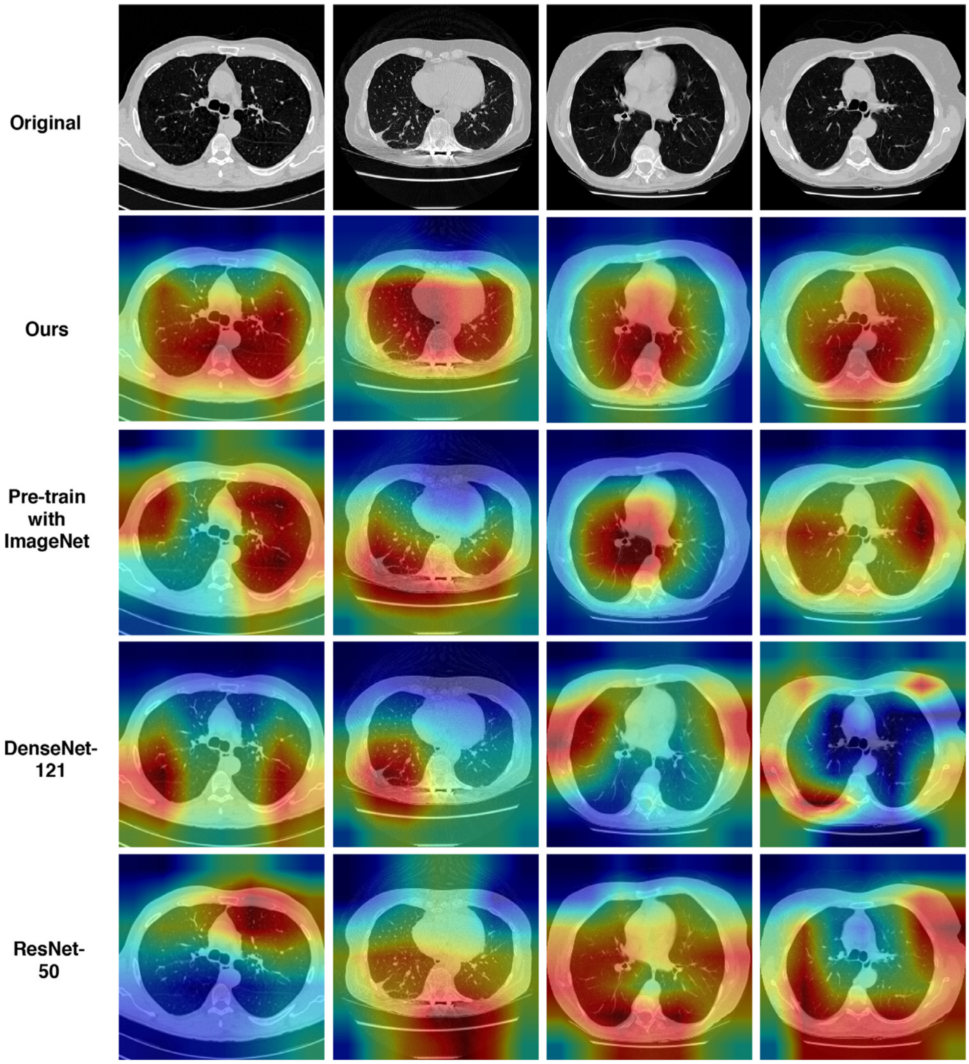

Fig. 3. Grad-CAM [42] visualizations of the features learned by different methods. The top row shows the original image set, followed by our method, pre-train with ImageNet and DenseNet-121 and ResNet-50. The results indicate that our method performs better than other approaches in learning lung features.

Table 4
Training time taken by different methods on our server (in hour(h)).

	COVID-19 + MegSeg	LIDC-IDRI + DeepLesion
ResNet-50	1	6
DenseNet-121	1.5	7
Ours	–	8

Table 5
The effect of data augmentation and projection head.

	Accuracy	Precision	Recall	AUC
Ours	0.868 ± 0.012	0.883 ± 0.011	0.872 ± 0.012	0.931 ± 0.013
No Aug.	0.779 ± 0.021	0.791 ± 0.020	0.780 ± 0.021	0.889 ± 0.022
No Proj.	0.856 ± 0.013	0.875 ± 0.012	0.870 ± 0.013	0.910 ± 0.015

pared data augmentation methods are shown in Fig. 2. In Table 6, we summarize the classification results and use AB-cropping, CD-cropping, and EF-cropping to represent each of the three strategies. The results demonstrate that the model performance decreases by varying degrees when using only one of the cropping strategies compared with using all of them. This suggests that combining all three cropping strategies yields better generalizability.

Table 6
Effects of using different augmentation strategies on classification performance.

	Accuracy	Precision	Recall	AUC
Random	0.868 ± 0.012	0.883 ± 0.011	0.872 ± 0.012	0.931 ± 0.013
AB	0.792 ± 0.011	0.801 ± 0.010	0.807 ± 0.010	0.844 ± 0.009
CD	0.821 ± 0.022	0.844 ± 0.014	0.832 ± 0.015	0.845 ± 0.014
EF	0.844 ± 0.018	0.852 ± 0.010	0.849 ± 0.013	0.866 ± 0.010

Table 7
Effect of fine-tuning.

	Accuracy	Precision	Recall	AUC
Ours	0.868 ± 0.012	0.883 ± 0.011	0.872 ± 0.012	0.931 ± 0.013
Linear	0.788 ± 0.011	0.795 ± 0.010	0.790 ± 0.010	0.892 ± 0.009

Table 8
Effects of resizing in data augmentation on model performance.

	Accuracy	Precision	Recall	AUC
With	0.868 ± 0.012	0.883 ± 0.011	0.872 ± 0.012	0.931 ± 0.013
Without	0.870 ± 0.013	0.885 ± 0.013	0.874 ± 0.011	0.932 ± 0.009

Table 9
Effects of using different encoders on model performance.

	Accuracy	Precision	Recall	AUC
ResNet-50	0.868 ± 0.012	0.883 ± 0.011	0.872 ± 0.012	0.931 ± 0.013
ResNet-152	0.861 ± 0.011	0.867 ± 0.013	0.862 ± 0.012	0.925 ± 0.012
DenseNet-161	0.861 ± 0.011	0.867 ± 0.013	0.862 ± 0.012	0.925 ± 0.012
VGG-16	0.849 ± 0.018	0.862 ± 0.010	0.859 ± 0.011	0.916 ± 0.011

Moreover, we also examined the effects of the fine-tuning process on the model performance. To do so, we first pre-trained the embedding network and modified the few-shot classification stage by replacing the prototypical network with a linear classifier with frozen features. We directly applied the linear classifier into the learned embedding network without any update on weights. The results are summarized in Table 7 which show that the fine-tuning process can significantly improve the performance.

In addition, we also investigated the impact of the resizing operation during the data augmentation process. To this end, we compared the model performance with and without the resizing operation (see Table 8). As can be seen, the resizing operation slightly affected the performance. Moreover, as the cropping operation may generate different-sized images, the resizing operation is necessary to ensure that all these generated images can be fed into the neural network for model training.

Finally, we examined the effect of using different encoders on model performance. We used the same settings as mentioned in Section 4.2, but changed the encoder network from ResNet-50 to ResNet-152, DenseNet-161, and VGG-16 for performance comparison. The results reported in Table 9 show that ResNet-50 achieved the best performance. This justified the use of ResNet-50 as the encoder in our proposed model.

5. Conclusion

CT imaging is attracting increasing attention as a screening tool for COVID-19. It provides visualization for monitoring disease progression and can help to evaluate the severity. However, the lack of annotated CT scans is a significant challenge in CT imaging-based medical studies. In this work, we proposed a new deep-learning based method that can be used for the automatic diagnosis of COVID-19 with limited samples. Moreover, we demonstrated that our method achieved superior performance over ResNet-50 when the number of available samples is larger than three. ResNet-50 is a well-known and widely used supervised learning model for medical image analysis. As our developed model used a self-supervised strategy based on unsupervised learning, the fact that it can outperform than ResNet-50 is remarkable.

We expect that our method will be useful for other medical imaging analysis tasks facing the same data shortage problem. In the future, we plan to apply the proposed method to more COVID-19 datasets to validate its generalizability. Moreover, we will also investigate how to use knowledge distillation to reduce the size of learned embedding and further increase the classification accuracy.

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.

References

[1] D.-P. Fan, T. Zhou, G.-P. Ji, Y. Zhou, G. Chen, H. Fu, J. Shen, L. Shao, INF-Net: automatic COVID-19 lung infection segmentation from CT images, IEEE Trans. Med. Imaging 39(8) (2020) 2626–2637, doi:10.1109/TMI.2020.2996645.

[2] A. Narin, C. Kaya, Z. Pamuk, Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks, arXiv:2003.10849(2020).

[3] Z. Wang, Y. Xiao, Y. Li, J. Zhang, F. Lu, M. Hou, X. Liu, Automatically discriminating and localizing COVID-19from community-acquired pneumonia on chest X-rays, Pattern Recognit. 110 (2020) 107613.

[4] C. Long, H. Xu, Q. Shen, X. Zhang, B. Fan, C. Wang, B. Zeng, Z. Li, X. Li, H. Li, Diagnosis of the coronavirus disease (COVID-19): RRT-PCR or CT? Eur. J. Radiol. 126 (2020) 108961.

[5] T. Ai, Z. Yang, H. Hou, C. Zhan, C. Chen, W. Lv, Q. Tao, Z. Sun, L. Xia, Correlation of chest CT and RT-PCR testing in coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases, Radiology 296 (2020) 200642.

[6] A. Bernheim, X. Mei, M. Huang, Y. Yang, Z.A. Fayad, N. Zhang, K. Diao, B. Lin, X. Zhu, K. Li, et al., Chest CT findings in coronavirus disease-19 (COVID-19): relationship to duration of infection, Radiology 295 (2020) 200463.

[7] Y. Li, L. Xia, Coronavirus disease 2019 (COVID-19): role of chest CT in diagnosis and management, Am. J. Roentgenol. 214 (2020) 1–7.

[8] S. Salehi, A. Abedi, S. Balakrishnan, A. Gholamrezanezhad, Coronavirus disease 2019 (COVID-19): a systematic review of imaging findings in 919 patients, Am. J. Roentgenol. 215 (2020) 1–7.

[9] C. Huang, Y. Wang, X. Li, L. Ren, J. Zhao, Y. Hu, L. Zhang, G. Fan, J. Xu, X. Gu, et al., Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China, Lancet 395 (10223) (2020) 497–506.

[10] L.-s. Wang, Y.-r. Wang, D.-w. Ye, Q.-q. Liu, A review of the 2019 novel coronavirus (COVID-19) based on current evidence, Int. J. Antimicrob. Agents 55 (2020) 105948.

[11] L. Li, L. Qin, Z. Xu, Y. Yin, X. Wang, B. Kong, J. Bai, Y. Lu, Z. Fang, Q. Song, et al., Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT, Radiology 296 (2020) 200905.

[12] X. Mei, H.-C. Lee, K.-y. Diao, M. Huang, B. Lin, C. Liu, Z. Xie, Y. Ma, P.M. Robson, M. Chung, et al., Artificial intelligence–enabled rapid diagnosis of patients with COVID-19, Nat. Med. 26 (2020) 1–5.

[13] J. Chen, L. Wu, J. Zhang, L. Zhang, D. Gong, Y. Zhao, Q. Chen, S. Huang, M. Yang, X. Yang, et al., Deep learning-based model for detecting 2019 novel coronavirus pneumonia on high-resolution computed tomography, Sci. Rep. 10 (1) (2020) 1–11.

[14] X. Chen, L. Yao, Y. Zhang, Residual attention U-Net for automated multiclass segmentation of COVID-19 chest CT images, arXiv preprint arXiv:2004.05645(2020b).

[15] A. Zhao, G. Balakrishnan, F. Durand, J.V. Guttag, A.V. Dalca, Data augmentation using learned transformations for one-shot medical image segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 8543–8553.

[16] A. Oliveira, S. Pereira, C.A. Silva, Augmenting data when training a CNN for retinal vessel segmentation: how to warp? in: 2017 IEEE 5th Portuguese Meeting on Bioengineering (ENBENG), IEEE, 2017, pp. 1–4.

[17] F. Cen, X. Zhao, W. Li, G. Wang, Deep feature augmentation for occluded image classification, Pattern Recognit. 111 (2020) 107737.

[18] W. Li, L. Fan, Z. Wang, C. Ma, X. Cui, Tackling mode collapse in multi-generator GANs with orthogonal vectors, Pattern Recognit. 110 (2020) 107646.

[19] Z. Eaton-Rosen, F. Bragman, S. Ourselin, M.J. Cardoso, Improving data augmentation for medical image segmentation, in: 1st Conference on Medical Imaging with Deep Learning, 2018.

[20] X. He, X. Yang, S. Zhang, J. Zhao, Y. Zhang, E. Xing, P. Xie, Sample-efficient deep learning for COVID-19 diagnosis based on CT scans, medRxiv (2020).

[21] Z. Fang, J. Ren, S. Marshall, H. Zhao, S. Wang, X. Li, Topological optimization of the densenet with pretrained-weights inheritance and genetic channel selection, Pattern Recognit. 109 (2020) 107608.

[22] A. van den Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, arXiv preprint arXiv:1807.03748(2018).

[23] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9729–9738.
[24] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, in: Advances in Neural Information Processing Systems, 2017, pp. 4077–4087.
[25] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., Matching networks for one shot learning, in: Advances in Neural Information Processing Systems, 2016, pp. 3630–3638.
[26] J. Donahue, K. Simonyan, Large scale adversarial representation learning, in: Advances in Neural Information Processing Systems, 2019, pp. 10541–10551.
[27] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, Decaf: a deep convolutional activation feature for generic visual recognition, in: International Conference on Machine Learning, 2014, pp. 647–655.
[28] R.D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, Y. Bengio, Learning deep representations by mutual information estimation and maximization, in: International Conference on Learning Representations, 2019.
[29] P. Bachman, R.D. Hjelm, W. Buchwalter, Learning representations by maximizing mutual information across views, in: Advances in Neural Information Processing Systems, 2019, pp. 15509–15519.
[30] O.J. Hénaff, A. Srinivas, J. De Fauw, A. Razavi, C. Doersch, S. Eslami, A. van den Oord, Data-efficient image recognition with contrastive predictive coding, arXiv preprint arXiv:1905.09272(2019).
[31] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: an imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, 2019, pp. 8024–8035.
[32] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), 2, IEEE, 2006, pp. 1735–1742.
[33] T. Chen, S. Kornblith, M. Norouzi, G.E. Hinton, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning, 2020.
[34] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[35] Z. Wu, Y. Xiong, S.X. Yu, D. Lin, Unsupervised feature learning via non-parametric instance discrimination, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3733–3742.
[36] C. Doersch, A. Zisserman, Multi-task self-supervised visual learning, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2051–2060.
[37] F. Schroff, D. Kalenichenko, J. Philbin, FaceNet: a unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.
[38] W. Wu, D. Tao, H. Li, Z. Yang, J. Cheng, Deep features for person re-identification on metric learning, Pattern Recognit. 110 (2020) 107424.
[39] Z. Cheng, X. Zhu, S. Gong, Face re-identification challenge: are face recognition models good enough? Pattern Recognit. 107 (2020) 107422.
[40] K. Yan, X. Wang, L. Lu, L. Zhang, A.P. Harrison, M. Bagheri, R.M. Summers, Deep lesion graphs in the wild: relationship learning and organization of significant radiology image findings in a diverse large-scale lesion database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9261–9270.
[41] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.
[42] 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, 2017, pp. 618–626.

Xiaocong Chen is a Ph.D. student at the School of Computer Science and Engineering, University of New South Wales. His research interests include recommendation system, machine learning, deep learning, and its applications.

Lina Yao received her Ph.D. from the University of Adelaide in 2014. She is a senior lecturer at the School of Computer Science and Engineering, University of New South Wales. Her research interests include data mining and machine learning applications, with a focus on IoT analytics, and Brain-Computer Interface.

Tao Zhou received Ph.D. degree in Pattern Recognition and Intelligent System from the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, in 2016. He is currently a Research Scientist at Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates. His research interests include machine learning, computer vision and medical image analysis. He is the Associate Editor of IEEE Access.

Jinming Dong is a bachelor student at the School of Computer Science and Engineering, University of New South Wales. His research focuses on Machine Learning, Deep Learning, Computer Vision and Natural Language Processing.

Yu Zhang is an Assistant Professor of Bioengineering at Lehigh University. He received postdoctoral training at the Department of Psychiatry and Behavior Sciences, Stanford University, and the Biomedical Research Imaging Center, University of North Carolina at Chapel Hill. He is the author of over 100 peer-reviewed papers that have been published in prestigious Journals, such as Nature Biomedical Engineering, Nature Biotechnology, Nature Human Behaviour, Proceedings of the IEEE, IEEE TCYB, IEEE TNNLS, IEEE TNSRE, and IEEE TBME. His research interests include computational neuroscience, brain network, pattern recognition, machine learning, signal processing, artificial intelligence, brain-computer interface, medical imaging computing.