

Improving COVID-19 CT Classification of CNNs by Learning Parameter-Efficient Representation

Yujia Xu^a, Hak-Keung Lam^{a,*}, Guangyu Jia^a, Jian Jiang^a, Junkai Liao^a and Xinqi Bao^a

^aDepartment of Engineering, King's College London, Strand, London, WC2R 2LS, United Kingdom

ARTICLE INFO

Keywords:

COVID-19
Computed Tomography
CNNs
Deep Learning
Similarity Regularization

ABSTRACT

COVID-19 pandemic continues to spread rapidly over the world and causes a tremendous crisis in global human health and the economy. Its early detection and diagnosis are crucial for controlling the further spread. Many deep learning-based methods have been proposed to assist clinicians in automatic COVID-19 diagnosis based on computed tomography imaging. However, challenges still remain, including low data diversity in existing datasets, and unsatisfied detection resulted from insufficient accuracy and sensitivity of deep learning models. To enhance the data diversity, we design augmentation techniques of incremental levels and apply them to the largest open-access benchmark dataset, COVIDx CT-2A. Meanwhile, similarity regularization (SR) derived from contrastive learning is proposed in this study to enable CNNs to learn more parameter-efficient representations, thus improving the accuracy and sensitivity of CNNs. The results on seven commonly used CNNs demonstrate that CNN performance can be improved stably through applying the designed augmentation and SR techniques. In particular, DenseNet121 with SR achieves an average test accuracy of 99.44% in three trials for three-category classification, including normal, non-COVID-19 pneumonia, and COVID-19 pneumonia. And the achieved precision, sensitivity, and specificity for the COVID-19 pneumonia category are 98.40%, 99.59%, and 99.50%, respectively. These statistics suggest that our method has surpassed the existing state-of-the-art methods on the COVIDx CT-2A dataset.

1. Introduction

The Coronavirus Disease 2019 (COVID-19) has become a worldwide pandemic and infected over 493 million people till April 2022 [1]. Its increasingly high infectivity and fatality rate due to strain variation are threatening human health and damaging the global economy [2–4]. The efficient reproductive number of the virus in many countries remains high, as reported in [5], indicating COVID-19 continues spreading quickly around the world. Thereby, a timely and efficient diagnosis is crucial for the treatment of COVID-19 positive patients and the control of further disease spread.

In the early diagnosis of COVID-19 infection, real-time reverse transcription polymerase chain reaction (RT-PCR) is the primary choice due to its convenience and high specificity. However, research results [6–8] have suggested that RT-PCR is not sensitive enough that some infected patients turned out to be positive even after several negative tests. These false-negative cases might continue to infect their close contacts without isolation or develop into severe illness. Chest computed tomography (CT) is a supplementary screening tool to RT-PCR since CT has higher sensitivity in detecting infection, indicated by institutes [9–11]. The high cost and hours of scanning time of CT are not affordable for all institutes. Thus CT can be more suitable in some scenarios where patients have suspicious negative RT-PCR tests, or patients are in need of timely diagnosis, or the RT-PCR test kits are undersupplied.

Since the pandemic started, researchers have been exploring the potential of convolutional neural networks

(CNNs) in COVID-19 CT classification and reported high accuracy without clinician intervention. CNNs are a kind of deep learning technique dominating computer vision tasks. For example, Gunraj *et al.* [12] introduced a large-scale open-access COVID-19 CT dataset (COVIDx CT-1) and trained a COVID-19-specific tailored CNN on it. Panwar *et al.* [13] utilized transfer learning to inherit cross-domain knowledge to improve the model performance. All these researches reveal that CNNs have the potential to serve as an assistant to help clinicians in COVID CT diagnosis.

Although CNNs have achieved remarkable results in CT diagnosis, challenges remain before they can be put into practical use. Deep learning methods often require large-scale standard datasets, while the existing COVID-19 CT datasets are insufficient. Also, CT scans collected from different institutes have inconsistent characteristics like orientation, brightness, etc. The trained models might be more sensitive to these irrelevant information rather than the pneumonic pathologies that really matter. Furthermore, the increasingly great capability of CNN-based models may not be fully fulfilled given the limited data sources. Hence, methods for learning more parameter-efficient representations are crucial for mitigating the data insufficiency issue and improving the classification performance.

By addressing these problems above, a more reliable COVID-19 CT classification system can reduce the workload of clinicians and provide more accurate and sensitive computer-aided diagnoses. Motivated by these factors, this study aims to use deep learning techniques to improve the COVID-19 CT classification performance of commonly used CNNs. Particularly, to alleviate the data insufficiency

Email addresses: {yujia.xu, hak-keung.lam, guangyu.jia, jian.jiang, junkai.liao, xinqi.bao}@kcl.ac.uk

*Corresponding author

and enhance the data diversity, we design and apply augmentation of incremental levels on the currently largest COVID-19 CT benchmark dataset (COVIDx CT-2A) [14]. Meanwhile, to find the optimal selection of CNN architectures and augmentation combinations, we explore seven commonly used CNN architectures under seven augmentation settings. The CNNs include SqueezeNet1.1 [15], MobileNetV2 [16], DenseNet121 [17], ResNet-18/34/50 [18], and InceptionV3 [19]. Meanwhile, contrastive learning is one promising self-supervised method for enabling deep learning models to learn more parameter-efficient features. We propose the similarity regularization (SR) derived from contrastive learning to learn more parameter-efficient representations and improve CNN classification. The experimental results demonstrate that SR can improve the classification performance of CNNs stably and surpass conventional contrastive learning. Our main contributions are summarized as follows:

- a) We investigate the impacts of augmentation and model selection in COVID-19 CT classification for three classes, including normal, non-COVID-19 pneumonia (NCP), and COVID-19 pneumonia (CP).
- b) We propose SR as a regularization term for learning more parameter-efficient representations. Comparisons between seven models with or without SR are conducted. The experimental results demonstrate that SR can improve classification stably without extra introduced model parameters during the test interface.
- c) Our proposed model DenseNet121-SR achieves 99.44% test accuracy, and 98.40% precision, 99.59% sensitivity and 99.50% specificity for COVID-19 positive class, achieving the state-of-the-art.
- d) On other COVID-19 CT datasets, i.e., SARS-CoV2 and COVIDx CT-1, our DenseNet121-SR outperforms the existing methods in terms of efficiency and accuracy.
- e) We extend the study to seven classic natural datasets and find that our DenseNet121-SR is superior to the original DenseNet121 for all tasks, indicating that our method can be generalized to general classification problems.

2. Related Works

2.1. COVID-19 Related Researches

CNNs are increasingly improving the COVID-19 CT classification with advanced algorithms and enhanced datasets. Numerous CNN-based methods achieving high accuracy have been proposed, indicating the potential of CNNs in assisting practical diagnosis. Some representative methods on four benchmark datasets are listed in Table 1.

In the COVID-19 CT classification, there exist no golden standard datasets so far. The four widely employed open-access datasets [12, 14, 20, 27] in Table 1 differ in many aspects, including patient/scan distribution, collection sources, dataset size, class numbers, labelling quality, etc. Particularly, COVID-CT [27] and SARS-CoV-2 [20] are two small binary-classification datasets containing 812 and 2,482 CT scans for COVID-19 positive and non-COVID classes, respectively. Gunraj *et al.* released a larger dataset COVIDx

CT-1 [12] consisting of 104,009 scans for normal, NCP and CP classes upon which the authors later built COVIDx CT-2 [14]. COVIDx CT-2 is the largest existing dataset containing 194,922 CT scans, combined from multiple data sources. Generally, data-driven methods like CNNs depends heavily on dataset size. This can be drawn from the classification metrics in Table 1 that the methods trained on larger datasets can roughly achieve higher performance. To ensure both the data diversity and satisfactory results, our study employs COVIDx CT-2A [14] as our target dataset.

Despite the various datasets, many CNN-based methods have been developed to continuously boost the classification performance. In particular, researchers often use transfer learning [13, 21–25, 33] or ensemble learning [22, 23, 25, 33, 34] to overcome the data insufficiency in small-scale datasets like SARS-CoV-2 and COVID-CT. For example, Jaiswal *et al.* [24] and Panwar *et al.* [13] utilized transfer learning to pre-train the weights of VGG19 and DenseNet201 on ImageNet and then fine-tuned on SARS-CoV-2, achieving 96.25% and 94.04% accuracy, respectively. Besides, ensemble learning, merging the decisions from multiple models into a more balanced decision, has been widely applied in some works through different merging approaches like weighted sum [25], voting [22], and fuzzy rank-based fusion [23]. However, ensemble learning is rarely applied in large-scale datasets like COVIDx CT-1/2 [12, 14]. Specifically, COVID-Net CT-1/2 L [12, 14] are two light-tailored CNNs whose architectures are finely designed by automatic neural architecture searching. The two models are extremely parameter-efficient and achieved 94.5% and 98.1% accuracy with only 0.45MB and 1.40MB parameters, respectively. Another research [32] employed ResNet-50x1 pre-trained on ImageNet-21k and fine-tuned on COVIDx CT-2A, achieving 99.2% accuracy.

Drawn from the reviewed research works above, deep learning models can achieve higher performance in COVID-19 CT classification through the approaches that: 1) train models over data of higher diversity; 2) with finely designed neural networks; 3) ensemble the decisions from multiple models; 4) inherit out-of-domain classification knowledge. Although models can benefit from these aspects, the expensive computational cost of neural architecture searching and large-scale pre-training, and long execution time caused by over-parameterization should be considered as well.

2.2. Contrastive Learning

In recent years, supervised deep learning models of increasing complexity and depth have shown great progress in many large-scale applications like ImageNet classification [18, 19]. However, directly applying these models to COVID-19 datasets of smaller scales might cause over-parameterization. It means that model capacities cannot be fully fulfilled and the extracted representations are not parameter-efficient. One promising approach to addressing the issue is contrastive learning.

In the deep learning field, it is widely recognized that the model performance depends on the quality of their

Table 1

Comparison of classification metrics between multiple deep learning methods in four datasets. The precision, sensitivity, and specificity metrics are for COVID-19 positive class only. (The decimal places are kept consistent as reported in the publications.)

Dataset	Method	Params. (M)	Accuracy (%)	Precision (%)	Sensitivity (%)	Specificity (%)
SARS-CoV-2 [20]	Alshazly <i>et al.</i> [21]	86.74	99.4	99.6	99.8	99.6
	Silva <i>et al.</i> [22]	4.78	98.99	99.20	98.80	-
	Kundu <i>et al.</i> [23]	132.86	98.93	98.93	98.93	98.93
	Jaiswal <i>et al.</i> [24]	-	96.25	96.29	96.29	96.21
	Panwar <i>et al.</i> [13]	20.55	94.04	95.30	94.04	95.86
	Jangam <i>et al.</i> [25]	202.87	93.5	89.91	98	-
	Wang <i>et al.</i> [26]	-	90.83	95.75	85.89	-
COVID-CT [27]	Chen <i>et al.</i> [28]	-	88.5	89.9	88.6	-
	He <i>et al.</i> [29]	0.55	86	-	-	-
	Polsinelli <i>et al.</i> [30]	1.26	85.03	85.01	87.55	81.95
	Jangam <i>et al.</i> [25]	202.87	84.73	78.15	94.9	-
	Wang <i>et al.</i> [26]	-	78.69	78.02	79.71	-
COVIDx CT-1 [12]	Gunraj <i>et al.</i> [12]	1.40	99.1	99.7	97.3	99.9
	Ter-Sarkisov [31]	34.14	91.66	90.80	94.75	-
COVIDx CT-2A [14]	Zhao <i>et al.</i> [32]	23.51	99.2	98.5	98.7	99.5
	Gunraj <i>et al.</i> [14]	0.45	98.1	97.2	98.2	98.8
	Gunraj <i>et al.</i> [12]	1.40	94.5	90.2	99.0	95.7

learned representation. Contrastive learning, also known as contrastive self-supervised representation learning, is one framework aiming at learning efficient representations without human-specified labels. In general, the main idea of contrastive learning is to project inputs into an embedding space where the embedded vectors of similar samples are closer while dissimilar ones are apart. More formally, for visual tasks, a pair of views augmented from one image is considered as a positive pair while pairs of views from different images are considered as negative pairs. Hence, contrastive learning models aim to maximize the representation similarity between positive pairs and minimize that between negative pairs. In practical tasks, contrastive learning often pre-trains the front representation extractors of deep learning models in a self-supervised manner, and then fine-tunes the pre-trained weights in a conventional supervised manner.

The state-of-the-art contrastive learning frameworks include MoCo [35, 36], SimCLR [37, 38], SimSiam [39], SwAV [40], BYOL [41], etc. These frameworks mainly differ in terms of loss function, representation projection, and negative pair formation [39]. And the differences further determine their requirements on complexity of augmentation policies and batch size. Normally, in order to obtain satisfactory result, contrastive methods depends on a large batch size to cover enough negative pairs [35–38]. Among these models, BYOL, SwAV and SimSiam are the contrastive frameworks requiring no negative pairs. In ImageNet linear classification experiments [39], BYOL achieves relatively better performance. This explains that we select BYOL as the basic framework for SR calculation as in Section 3.2.

The success of contrastive learning has emerged some applications in COVID-19 CT diagnosis [28, 29, 42]. He *et al.* [29] employed a MoCo-like [35] framework to enhance

the CT scans representations extracted by DenseNet169 and fine-tuned the network, achieving 86% accuracy on COVID-CT [27]. Similarly, Chen *et al.* employed the MoCo-v2-like [36] framework on the same dataset and reached 88.5% accuracy within six shots. Li *et al.* [42] put the contrastive loss as a regularization term and trained their CMT-CNN in an end-to-end manner, obtaining 93.46% accuracy. These studies suggest contrastive learning can boost classification performance by learning more efficient representations.

3. Method

3.1. Augmentation of Incremental Levels

Data augmentation is vital for improving the performance of deep learning models, especially for contrastive learning [36, 37]. However, the optimal selection for COVID-19 CT augmentation has not been studied. Inspired by the literature in Section 2, we design and evaluate a series of augmentation operations of incremental levels as follows where "+" denotes the appended augmentation based on the previous level: **Level 0**: No augmentation; **Level 1** + RandomResizedCrop: Randomly obtain an image crop of size in the range [0.08, 1] of the original size 256×256 , and then randomly scale the crop according to an aspect ratio in the range [3/4, 4/3]. The scaled crop is finally resized to the original size; **Level 2** + Horizontal flip: Randomly flip the input image horizontally with 50% using probability; **Level 3** + RandAugment [43]: Randomly apply rand augment twice with magnitude 9 and magnitude standard deviation 0.5; **Level 4** + Random Erasing [44]: Select a rectangle region of the input image and do pixel-wise erasing with 25% using probability. The size of the selected region are randomly picked in the range [0.02, 1/3] of the image size. **Level 5** + Mixup [45]: Mix two in-batch images up with a ratio λ

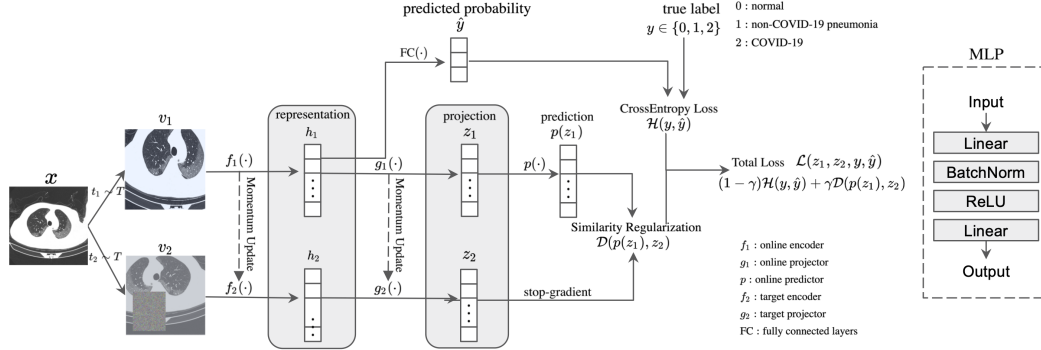


Figure 1: The overall structure of the models with our proposed similarity regularization in training interface. The two projectors, g_1 and g_2 , and the online predictor p are implemented by non-linear MLPs. After training, only the online encoder f_1 and fully connected layer FC are preserved in testing.

subjecting to a beta distribution, $\lambda \sim B(1, 1)$. The mixup process for images I_A and I_B can be formatted as $I_A(x, y) = \lambda I_A(x, y) + (1 - \lambda) I_B(x, y)$, where (x, y) denotes the pixel coordinate; **Level 6 + CutMix** [46]: Switch from Mixup to Cutmix with 50% probability. Randomly replace a square region in the original image with a region in another in-batch image. The region size is randomly determined, subject to the squared root of a beta distribution $B(1, 1)$.

The visualization of the augmented scans is demonstrated in Fig. 4 in Appendix A. Specifically, RandomResizedCrop and horizontal flip are two commonly used augmentation operations in both supervised [18, 19] and self-supervised learning [36–38, 41]. Since contrastive learning requires more complicated augmentation [37], the two stronger augmentations, RandAugment and Random Erasing, are further introduced in levels 3 and 4. Their implementations and parameters refer to [47, 48]. In levels 5 and 6, mixup and cutmix are two augmentations enabling higher data diversity by fusing in-batch images. In these two levels, we mainly experiment on whether such sample-fusing augmentations can improve COVID-19 classification. By comparing the performance of models under these incremental augmentation levels, an appropriate augmentation strategy for COVID-19 CT scans can be established.

3.2. Similarity Regularization

Most mainstream conventional CNNs contains two parts, a representation extractor f and a followed fully connected layer FC. The extractor f aims to extract the distinguishable representations of given inputs, and FC predicts the class probability distribution by summarizing the extracted representations. This forward propagation is demonstrated as the top branch in Fig. 1. More formally, the input image x is first transformed to a view v by a random on-the-fly augmentation operation $t \sim T$ where T denotes an infinite collection of augmentation operations. Subsequently, the representation extractor f converts the input view v to a representation embedding vector $h = f(v)$. The followed FC predicts the class probability distribution based on its obtained representation, $\hat{y} = FC(h)$. The training target of such a classifier is to minimize the class

probability distribution distance between the prediction \hat{y} and the ground truth y according to the cross-entropy loss in Eq. (1), where $i \in \{0, 1, 2\}$ denotes the class index.

$$\mathcal{H}(y, \hat{y}) = - \sum_{i=0}^2 y_i \log \hat{y}_i \quad (1)$$

In this conventional fully supervised scenario, the trained representations aim at better projecting to human-specified class distribution. However, this manner affects data efficiency, robustness or generalization [49]. Instead, contrastive learning enables learning more parameter-efficient representations from inputs themselves instead of the specified annotations. We thus incorporate it in common CNNs to improve their representation learning ability.

The overall structure of our method is illustrated in Fig. 1. We keep the conventional supervised classifier unchanged in the top branch while introducing a contrastive learning framework in the bottom branch. As in Section 2.2, contrastive learning is to maximize the representation similarity between positive pairs. We punish the positive-pair representation distance as a regularization term beside the cross-entropy loss, naming the term similarity regularization (SR).

Particularly, the contrastive framework is a siamese network like most mainstream frameworks [37–39, 41], consisting of an online network and a target network. The target network can be seen as a moving average of the online one. Given two views v_1 and v_2 augmented from the same input image x , the representation extractors f_1 and f_2 in two networks extract their corresponding latent representation vectors, $h_1 = f_1(v_1)$ and $h_2 = f_2(v_2)$. To avoid representations heavily affected by SR, the representation vectors then projected to another embedding space where $z_1 = g_1(h_1)$ and $z_2 = g_2(h_2)$, as in [38, 41]. Since the projectors g_1 and g_2 share slightly different feature spaces, the online projection z_1 is further projected to $p(z_1)$ of same dimension via online predictor p . The cosine representation similarity \mathcal{S} of value in range $[-1, 1]$ can be measured according to Eq. (2).

$$\mathcal{S}(p(z_1), z_2) = \frac{\langle p(z_1), z_2 \rangle}{\|p(z_1)\|_2 \cdot \|z_2\|_2} \quad (2)$$

where $\langle \cdot, \cdot \rangle$ and $\|\cdot\|_2$ are inner product and L^2 norm notations, respectively. A higher value indicates two vectors are of higher similarity. To penalize a low cosine similarity between positive pairs and scale the penalty in range $[0, 1]$, SR can be calculated as Eq. (3).

$$D(p(z_1), z_2) = 2 - 2S(p(z_1), z_2) = 2 - \frac{2\langle p(z_1), z_2 \rangle}{\|p(z_1)\|_2 \cdot \|z_2\|_2} \quad (3)$$

Hence, for a positive pair (v_1, v_2) , its total loss containing both cross-entropy loss and SR is written as in Eq. (4).

$$\mathcal{L}(z_1, z_2, y, \hat{y}) = (1 - \gamma)\mathcal{H}(y, \hat{y}) + \gamma D(p(z_1), z_2) \quad (4)$$

where γ is a scale factor for balancing the conventional cross-entropy loss and the introduced SR.

(v_2, v_1) is the symmetric positive pair with respect to (v_1, v_2) . We calculate the losses for both symmetric pairs and take their mean as the final loss for fast convergence.

SR as a regularization term may raise the concern if it will dominate the combined loss and thus degrade the classification. To remove the concern and find an appropriate scheduler for γ , we design three strategies as listed below. i denotes the current training iteration number.

Constant (default) : γ is set to be a constant value during all training iterations, 0.5 by default.

Linear Decay : γ decays linearly to a minimum value $\gamma_{min} = 0.01$ along N training iterations according to Eq. (5).

$$\gamma_i = \gamma_{min} + (1 - \frac{i}{N})(1 - \gamma_{min}) \quad (5)$$

Cosine Decay : γ decays to a minimum value $\gamma_{min} = 0.01$ along N training iterations according to cosine annealing scheduler as in Eq. (6).

$$\gamma_i = \gamma_{min} + \frac{1}{2}(1 + \cos \frac{i\pi}{N})(1 - \gamma_{min}) \quad (6)$$

Besides, after training, we throw away all the components except the online representation extractor f_1 and the fully connected layer FC. Hence, introducing SR in training will not slow down the test interface. The training pseudocode of models with SR is demonstrated in Algorithm 1.

4. Results and Analysis

4.1. Dataset Description

In this paper, we mainly train and evaluate our proposed method using the largest existing open-access COVID-19 CT dataset, COVIDx CT-2A². Specifically, the dataset contains three classes, including normal, non-COVID-19 pneumonia (NCP), and COVID-19 pneumonia (CP). Its class distribution is summarized in Table 2. The dataset is of high diversity, containing scans of 3,745 patients from eight open-access sources. It should be noted that the scans from the same patient are in one subset, preventing information leakage from training to validation or testing.

²<https://www.kaggle.com/hgunraj/covidxct>

Algorithm 1: Algorithm for training models with our proposed similarity regularization.

Inputs: batch size B ; iterations N ; network components $f_1, f_2, g_1, g_2, p, FC$; cross entropy criterion \mathcal{H} ; SR criterion D ; loss scale factor γ ; momentum rate β .

for $i = 1$ **to** N **do**

Sample a minibatch $\{(x_k, y_k)\}_{k=1}^B$;

for $k = 1$ **to** B **do**

// Augmentation and predict representation

$t_1 \sim T, t_2 \sim T$;

$v_1 = t_1(x_k), v_2 = t_2(x_k)$;

$h_1^{(1)} = f_1(v_1), h_2^{(1)} = f_2(v_2)$;

$h_1^{(2)} = f_1(v_2), h_2^{(2)} = f_2(v_1)$;

// Calculate cross-entropy

$\hat{y}^{(1)} = FC(h_1^{(1)}), \hat{y}^{(2)} = FC(h_1^{(2)})$;

$\mathcal{L}_k^{CE} = 0.5\mathcal{H}(y_k, \hat{y}^{(1)}) + 0.5\mathcal{H}(y_k, \hat{y}^{(2)})$;

// Calculate similarity regularization

$pz_1^{(1)} = p(g_1(h_1^{(1)})), z_2^{(1)} = g_2(h_1^{(1)})$;

$pz_1^{(2)} = p(g_1(h_1^{(2)})), z_2^{(2)} = g_2(h_1^{(2)})$;

Stop gradient for $z_2^{(1)}$ and $z_2^{(2)}$;

$\mathcal{L}_k^{SR} = 0.5D(pz_1^{(1)}, z_2^{(1)}) + 0.5D(pz_1^{(2)}, z_2^{(2)})$;

end

// Scale the batch loss and update networks

Set γ_i according to applied scheduler;

$\mathcal{L}_{total} = \frac{1}{B} \sum_{k=1}^B [(1 - \gamma_i)\mathcal{L}_k^{CE} + \gamma_i\mathcal{L}_k^{SR}]$;

Update f_1, g_1, p, FC by back-propagating \mathcal{L}_{total} ;

// Momentum update weights of f_2 and g_2

$f_2 = (1 - \beta)f_1 + \beta f_2$;

$g_2 = (1 - \beta)g_1 + \beta g_2$;

end

Return f_1 and FC ;

Table 2

Class distribution of the employed COVIDx CT-2A dataset.

Set	Normal	NCP	CP	Total
Training	35,996	25,496	82,286	143,778
Validation	11,842	7,400	6,244	25,486
Testing	12,245	7,395	6,018	25,658

4.2. Experimental Setting

In this paper, we keep the hyper-parameters consistent in all experiments for fair comparisons. The codes are implemented by PyTorch. We implement the CNN backbones and image augmentation by torchvision and timm [48] libraries, respectively. For acceleration, we train models on Torch distributed data parallelism on four Nvidia V100 GPUs with apex mixed precision of level O1. Besides, to alleviate the randomness concern, we obtain the experimental statistics by averaging the measurements in three distinct trials.

During training, CT scans are resized to 256×256 in 3 channels using bicubic interpolation and normalized by ImageNet mean and std. In the test interface, 256×256 CT scans are cropped from the center of resized 293×293 original images. This is empirically good as the center crop

can preserve the main lung regions. To avoid models from being too confident in one-class prediction, label smoothing [19] of smoothing factor 0.1 is applied in the cross-entropy in augmentation levels 0 – 4. While in augmentation level 5 or 6, in-batch paired labels are mixed up based on mixed inputs (See [45, 46] for more details).

The optimizer we used is Adam with 10^{-6} weight decay. After a 5-epoch linear warmup [50] from 5×10^{-7} , we use cosine annealing scheduler to decay the learning rate from 5×10^{-4} to 5×10^{-7} in the later 45 epochs. The batch size is set to be 64 in each process. Besides, the gradients are clipped to be no larger than 5.0 to avoid overflow.

In the SR calculation, the projectors g_1, g_2 and predictor p have the same multi-layer perceptron (MLP) architecture that consists of two linear layers connected by a batch normalization layer and a ReLU activation layer. The front linear layer projects the inputs to 512-D embedding vectors and the later linear layer outputs 128-D vectors. The analysis for the dimension setting is in Appendix B. The momentum rate β for updating f_2 and g_2 is 0.99, a median value among contrastive frameworks [35, 39–41].

4.3. Results of ResNets under Incremental Augmentation Levels

We first compare the performance of ResNets with or without SR under the incremental augmentation levels designed in Section 3.1 to determine an appropriate augmentation policy for the coming experiments. The averaged test accuracies are listed in Table 3. Since SR requires calculating the similarity between two augmented views, models with SR cannot be implemented under augmentation level 0.

Table 3

Test accuracy comparison between original ResNets and ResNets with proposed SR under incremental augmentation levels. ResNet is abbreviated as R. The scale factor scheduler for scaling SR is the default constant scheduler $\gamma = 0.5$.

Augmentation	Method	CNN Backbone		
		R18	R34	R50
Level 0	Original	91.28	92.28	91.57
Level 1	Original	99.10	99.00	98.89
	+SR(Ours)	99.23	99.27	99.09
Level 2	Original	99.17	99.11	99.00
	+SR(Ours)	99.27	99.39	99.19
Level 3	Original	99.11	99.06	99.08
	+SR(Ours)	99.26	99.29	99.20
Level 4	Original	99.12	99.09	99.13
	+SR(Ours)	99.40	99.43	99.31
Level 5	Original	97.49	97.34	97.66
Level 6	Original	98.50	98.69	98.91

Table 3 shows that the original ResNets achieve the highest accuracy in level 2 and cannot be improved in the following levels, heavily degraded in levels 5 and 6. The

degradation may result from the fact that sample-fusing augmentation sometimes transfers the pneumonic pathologies from CP/NCP cases to normal cases. We thus do not perform SR in levels 5 and 6. Different from the original ResNets, ResNets with SR continue to improve after level 2 and achieve the highest accuracy in level 4. This is consistent with the findings in many contrastive learning research works that contrastive learning requires stronger augmentation than supervised models [36, 37, 51]. Hence, we select level 4 as the basic augmentation level for the following experiments. Overall, it is observed that SR can improve the classification performance of ResNets stably under all augmentation levels from 1 to 4.

4.4. Results under Augmentation Level 4

In this section, we extend the experiments to seven widely used CNNs, including SqueezeNet1.1 [15], MobileNetV2 [16], DenseNet121 [17], ResNet-18/34/50 [18], and InceptionV3 [19]. The experiments are under augmentation level 4 and a constant scale factor scheduler $\gamma = 0.5$.

From the results in Table 4, it can be seen that all our models with SR surpass the original models in terms of averaged test accuracy. The best model, DenseNet121 with SR, achieves 99.44% accuracy with 7.33M parameters. Note that the extra parameters will be thrown away after training so that the parameters in the test interface are consistent for a model with or without SR. Another observation is that, in COVID-19 CT classification, the model performance is not in strictly proportional to its capacity despite of model architecture. This suggests that the fine design of model architecture rather than simply expanding depth or width is more valuable in this task, as supported in [12, 14, 30].

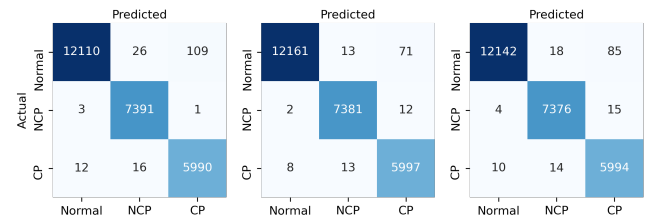


Figure 2: Confusion matrix for DenseNet121-SR in three trials.

Fig. 2 shows the confusion matrices for DenseNet121-SR in three training trials. Based on the matrices, we measure the performance of the model in terms of averaged accuracy, precision, sensitivity, and specificity, as listed in Table 5. The results show that our DenseNet121-SR has outperformed the state-of-the-art models in nearly all measurements. Specifically, DenseNet121-SR achieves a high sensitivity 99.59% for COVID-19 positive class, indicating that the model has the potential to efficiently avoid COVID-19 positive patients from being wrongly diagnosed.

Besides, to better understand the classification principles of the model, its attention visualized by Grad-CAM [52] is demonstrated in Fig. 6 in Appendix C.

Table 4

Test accuracy of seven CNNs with or without SR under augmentation level 4, ordered by the number of parameters. The scale factor scheduler for scaling SR is the default constant strategy $\gamma = 0.5$. Note that the extra parameters introduced by SR will be discarded in test interface after training.

Method	Metric	CNN Backbone						
		SqueezeNet	MobileNet	DenseNet	ResNet18	ResNet34	InceptionV3	ResNet50
Original	Params. (M)	0.69	2.12	6.63	10.66	20.30	20.78	22.42
	Acc. (%)	98.22	99.02	99.05	99.08	99.06	99.22	99.12
+SR(Ours)	Params. (M)	1.13	2.94	7.33	11.10	20.74	21.97	23.62
	Acc. (%)	98.41	99.18	99.44	99.39	99.43	99.32	99.31

Table 5

Comparison of DenseNet121-SR with the state-of-the-art methods on COVIDx CT-2A dataset.

Method	Accuracy (%)	Precision (%)			Sensitivity (%)			Specificity (%)		
		Normal	NCP	CP	Normal	NCP	CP	Normal	NCP	CP
COVID-Net CT-1 [12]	94.5	96.1	97.6	90.2	98.8	80.2	99.0	96.3	99.4	95.7
COVID-Net CT-2 S [14]	97.9	99.3	96.4	97.0	98.9	95.7	98.1	99.3	98.9	98.8
COVID-Net CT-2 L [14]	98.1	99.4	96.7	97.2	99.0	96.2	98.2	99.5	99.0	98.8
Bit-M [32]	99.2	99.8	98.9	98.5	99.3	99.6	98.7	99.8	99.6	99.5
DenseNet121-SR (Ours)	99.44	99.89	99.55	98.40	99.12	99.83	99.59	99.91	99.82	99.50

4.5. Results on Other Datasets

On other COVID-19 CT datasets Based on the experimental results aforementioned, we extend our method to the other two COVID-19 CT datasets, i.e., SARS-CoV2 and COVIDx-CT-1. It should be noted that, for SARS-CoV2, we train DenseNet121-SR over 200 epochs with weights pre-trained on ImageNet because SARS-CoV2 contains much fewer CT scans than the others. The results as listed in Table 6 show that our method can be generalized to other datasets and can achieve a high classification performance. Comparing to the methods listed in Table 1, our DenseNet121-SR with only 6.63 MB parameters is more parameter-efficient and outperforms the reviewed methods.

Table 6

Classification results of DenseNet121-SR on SARS-CoV2 and Covidx CT-1 datasets. The precision, sensitivity, and specificity metrics are for COVID-19 positive class only.

Dataset	Acc (%)	Prec (%)	Sens (%)	Spec (%)
SARS-CoV2	99.20	99.47	98.93	99.46
COVIDx CT-1	99.78	99.56	99.84	99.74

On classic natural datasets Besides, extensive experiments are conducted over seven natural datasets to further evaluate the generalization ability of our method. To evaluate the effect of SR fairly, we keep the setting unchanged as in Section 4.2 and initialize the model weights as pre-trained on ImageNet. Table 7 demonstrates the classification accuracy of DenseNet121 with or without SR on the seven datasets, including FGVC Aircraft [53], CIFAR10/100 [54], Describable Textures Dataset (DTD) [55], Oxford 102 Flowers [56], Oxford-IIIT Pets [57], and Stanford Cars [58].

It shows that DenseNet121-SR is superior to the original model in all the tasks, indicating our proposed SR can be generalized to general classification problems.

5. Ablation Study

The following ablation studies are conducted to better investigate the effects of our proposed SR.

Fully self-supervised learning Contrastive learning is widely adopted in pre-training CNNs that are fine-tuned later for downstream tasks. In our methods, we turn the process to an end-to-end manner by regularizing CNNs with proposed SR derived from contrastive learning. Hence, a comparison between SR and conventional contrastive learning is necessary. Specifically, we design and measure the following methods for comparison as follows.

- Linear Evaluation.** First pre-train the representation extractor f weights of which are frozen in the later FC fine-tuning. The pre-training process is equivalent to setting $\gamma = 1$ in all training epochs as in Algorithm 1, and then fine-tuning only the linear layers FC as usual. The fine-tuning hyper-parameters include: 256 batch size, learning rate decays from 40 to 4×10^{-6} according to cosine decay scheduler [50]. The optimizer used is SGD. Linear evaluation is simply conducted for verifying the effects of contrastive learning in this task.
- Two-stage training, self-supervised contrastive learning followed by conventional supervised learning.** Pre-training the representation extractor f and training the entire CNN with pre-trained weights. The hyper-parameters are consistent with others as in Section 4.2.
- Apply SR to ResNets with a default constant $\gamma = 0.5$.**

Table 7

Classification accuracy of DenseNet121 with or without SR over seven classic natural datasets.

	Aircraft	CIFAR10	CIFAR100	DTD	Flowers102	OxfordIIITPet	StanfordCars
DenseNet121	88.15	94.45	85.08	70.60	93.17	92.47	92.46
DenseNet121-SR (Ours)	88.18	94.47	85.08	71.01	94.42	92.88	92.55

Table 8

Test accuracy comparison between two-stage training and end-to-end training with SR under incremental augmentation levels. The scale factor scheduler for scaling SR is the default constant scheduler $\gamma = 0.5$.

Augmentation	Method	CNN Backbone		
		R18	R34	R50
Level 1	Linear Eval	95.47	92.89	92.32
	Two-stage	99.17	99.17	99.04
	+SR(Ours)	99.23	99.27	99.09
Level 2	Linear Eval	95.76	92.03	94.02
	Two-stage	99.25	99.23	99.20
	+SR(Ours)	99.27	99.39	99.19
Level 3	Linear Eval	93.92	93.88	95.08
	Two-stage	99.26	99.29	99.24
	+SR(Ours)	99.26	99.29	99.20
Level 4	Linear Eval	93.12	93.23	94.53
	Two-stage	99.38	99.30	99.18
	+SR(Ours)	99.40	99.43	99.31

The results for the designs above are listed in Table 8. It can be observed that contrastive learning can learn efficient representations that even a simple linear evaluation on the pre-trained representation extractor can achieve over 92% test accuracy. For the second method, two-stage contrastive learning, the pre-trained weights from the representation extractor might be hard to maintain in the later training phase. Our introduced SR maintains the representation by explicitly penalizing the representation difference for positive pairs. The results in Table 8 verify that ResNets with SR surpass the two-stage contrastive learning method in most experiments. Besides, it is worth noting that the end-to-end training with SR does not require pre-training and thus saves computational resources.

Decay strategy for γ The two-stage contrastive learning method can be approximate to run the SR Algorithm 1 with $\gamma = 1$ in pre-training and $\gamma = 0$ in fine-tuning. The sharp fall of γ may destroy the maintained representation space obtained in contrastive pre-training. To avoid the potential negative impact, we designed two mild γ decay strategies in Section 3.2 despite of the constant γ strategy. From the results demonstrated in Fig. 3, we can conclude that SR with all designed γ strategies can stably improve the classification accuracy. And SR is insensitive to the γ strategy setting since all strategies have comparable performance. Due to the simplicity of the γ strategy ($\gamma = 0.5$ in all iterations) and its

slight superiority in level 4 augmentation, we select it as the default strategy in our experiments.

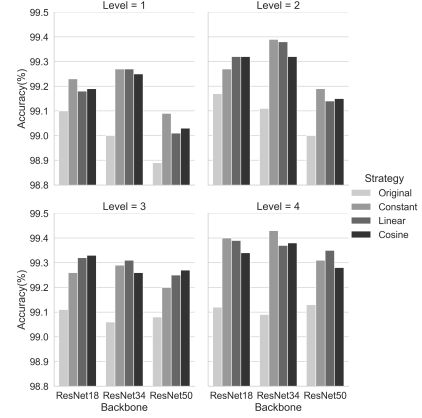


Figure 3: Accuracy of ResNets obtained with different γ decay strategies under incremental augmentation levels from 1 to 4. The baselines are the original models without introduced SR. The γ value for the constant strategy is 0.5 by default while γ decays from 1.0 to 0.01 in linear and cosine strategies.

γ value in constant strategy The ablation studies find that a constant strategy, $\gamma = 0.5$, can achieve the relatively highest performance among the three strategies under level 4 augmentation. We vary the constant γ value to evaluate its robustness. See Appendix D for the detailed results.

6. Discussion

Since COVID-19 grows rapidly worldwide, designing efficient and accurate classification systems is essential. Although some methods [12, 14, 21, 22, 32] have claimed a high classification accuracy ($\approx 99\%$) on multiple datasets, we argue that even a slight improvement can mitigate further infection. Meanwhile, some high-performance methods require considerable computational resources, making them hard to be deployed into practical healthcare systems. Hence, designing more efficient models with an affordable number of training parameters should also be noted.

In this paper, we propose an incremental augmentation strategy and SR to improve the CNN classification performance on three COVID-19 CT datasets. The results illustrate that appropriate augmentation can significantly alleviate the data limitation problem in COVID-19 CT classification. Meanwhile, our proposed SR further improves the classification performance of seven CNNs by enhancing their representation learning ability. Specifically, on the

largest dataset COVIDx CT-2A, our model DenseNet121-SR achieves 99.44% accuracy and 99.59% sensitivity with only 6.63 MB parameters in the test interface, outperforming all the reviewed state-of-the-art methods. Besides, we evaluate DenseNet121-SR on the other two datasets, achieving 99.78% and 99.20% accuracy on COVIDx CT-1 and SARS-CoV2, respectively. To further justify the effect of SR, we extend the DenseNet121-SR to seven classic natural datasets, illustrating SR can be generalized to general classification tasks. Furthermore, since SR derives from contrastive learning, we compare traditional contrastive learning and end-to-end training with SR in ablation studies. The comparison demonstrates that SR is superior in classification accuracy and training efficiency and is robust to its hyper-parameter setting.

Despite the achieved promising performance, the limitations of our method exist. Our method requires either large amounts of training data or pre-training on other large-scale datasets. The high performance of our models with SR partly owes to the efforts of workers collecting numerous CT scans. For smaller-scale datasets like SARS-CoV2, the backbone of our method requires pre-training. The pre-training on ImageNet helps improve the accuracy from around 98% to 99.20% in the DenseNet121-SR case. Besides, due to the lack of computational resources, we can hardly evaluate other contrastive frameworks. Meanwhile, we cannot redesign the CNN backbones to better balance computational efficiency and classification performance because of the substantial computational loads of neural architecture searching and pre-training. For future work, we will explore redesigning the network backbone, pre-training the redesigned backbones on large-scale datasets, and making networks more explainable in COVID-19 CT diagnosis.

7. Conclusion

This paper aims to improve the CNN performance for COVID-19 CT classification by enabling CNNs to learn parameter-efficient representations from CT scans. We propose the SR technique derived from contrastive learning and apply it to seven commonly used CNNs. The experimental results show that SR can stably improve the CNN classification performance. Together with a well-designed augmentation strategy, our model DenseNet121-SR with 6.63 MB parameters outperforms the existing methods on three COVID-19 CT datasets, including SARS-CoV2, COVIDx CT-1, and COVIDx CT-2A. Specifically, on the largest available dataset COVIDx CT-2A, DenseNet121-SR achieves 99.44% accuracy, with 98.40% precision, 99.59% sensitivity, and 99.50% specificity for the COVID-19 pneumonia category. Furthermore, the extensive experiments on seven classic natural datasets demonstrate that SR can be generalized to common classification problems.

CRedit authorship contribution statement

Yujia Xu: Conceptualization, Methodology, Writing original draft. **Hak-Keung Lam:** Method improvement, Review & editing, Supervision. **Guangyu Jia:** Method improvement. **Jian Jiang:** Review & editing. **Junkai Liao:** Review & editing. **Xinqi Bao:** Review & editing.

Acknowledgement

This work was supported by King's College London and has been performed using resources from the Cirrus UK National Tier-2 HPC Service at EPCC funded by the University of Edinburgh and EPSRC (EP/P020267/1).

References

- [1] WHO, WHO Coronavirus (COVID-19) dashboard, <https://covid19.who.int/>, 2022. [Online; Last accessed 7 April 2022].
- [2] W. McKibbin, R. Fernando, et al., The economic impact of COVID-19, *Economics in the Time of COVID-19* 45 (2020).
- [3] G. Iacobucci, COVID-19: New UK variant may be linked to increased death rate, early data indicate, *British Medical Journal* 372 (2021) n230.
- [4] E. Mahase, COVID-19: Where are we on vaccines and variants?, *British Medical Journal* 372 (2021) n597.
- [5] Y. Gu, COVID-19 infections tracker, <https://covid19-projections.com/infections-tracker/>, 2022. [Online; Last accessed 17 August 2021].
- [6] L. M. Kucirka, S. A. Lauer, O. Laeyendecker, D. Boon, J. Lessler, Variation in false-negative rate of reverse transcriptase polymerase chain reaction-based SARS-CoV-2 tests by time since exposure, *Annals of Internal Medicine* 173 (2020) 262–267.
- [7] Y. Li, L. Yao, J. Li, L. Chen, Y. Song, Z. Cai, C. Yang, Stability issues of RT-PCR testing of SARS-CoV-2 for hospitalized patients clinically diagnosed with COVID-19, *Journal of Medical Virology* 92 (2020) 903–908.
- [8] A. Tahamtan, A. Ardebili, Real-time RT-PCR in COVID-19 detection: issues affecting the results, *Expert Review of Molecular Diagnostics* 20 (2020) 453–454.
- [9] A. Kovács, P. Palásti, D. Veréb, B. Bozsik, A. Palkó, Z. T. Kincses, The sensitivity and specificity of chest CT in the diagnosis of COVID-19, *European Radiology* 31 (2021) 2819–2824.
- [10] Y. Fang, H. Zhang, J. Xie, M. Lin, L. Ying, P. Pang, W. Ji, Sensitivity of chest CT for COVID-19: comparison to RT-PCR, *Radiology* 296 (2020) E115–E117.
- [11] X. Xie, Z. Zhong, W. Zhao, C. Zheng, F. Wang, J. Liu, Chest CT for typical coronavirus disease 2019 (COVID-19) pneumonia: relationship to negative RT-PCR testing, *Radiology* 296 (2020) E41–E45.
- [12] H. Gunraj, L. Wang, A. Wong, COVIDNet-CT: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest CT images, *Frontiers in Medicine* 7 (2020) 1025.
- [13] H. Panwar, P. Gupta, M. K. Siddiqui, R. Morales-Menendez, P. Bhardwaj, V. Singh, A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest x-ray and CT-Scan images, *Chaos, Solitons & Fractals* 140 (2020) 110190.
- [14] H. Gunraj, A. Sabri, D. Koff, A. Wong, COVID-Net CT-2: Enhanced deep neural networks for detection of COVID-19 from chest CT images through bigger, more diverse learning, 2021. *arXiv:arXiv:2101.07433*.
- [15] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and 0.5 mb model size, *arXiv preprint arXiv:1602.07360* (2016).
- [16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, in: *Proceedings*

- of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.
- [17] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
 - [18] 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.
 - [19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.
 - [20] P. Angelov, E. Almeida Soares, SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification, MedRxiv (2020).
 - [21] H. Alshazly, C. Linse, E. Barth, T. Martinetz, Explainable COVID-19 detection using chest CT scans and deep learning, Sensors 21 (2021) 455.
 - [22] P. Silva, E. Luz, G. Silva, G. Moreira, R. Silva, D. Lucio, D. Menotti, COVID-19 detection in CT images with deep learning: A voting-based scheme and cross-datasets analysis, Informatics in Medicine Unlocked 20 (2020) 100427.
 - [23] R. Kundu, H. Basak, P. K. Singh, A. Ahmadian, M. Ferrara, R. Sarkar, Fuzzy rank-based fusion of CNN models using gompertz function for screening covid-19 ct-scans, Scientific reports 11 (2021) 1–12.
 - [24] A. Jaiswal, N. Gianchandani, D. Singh, V. Kumar, M. Kaur, Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning, Journal of Biomolecular Structure and Dynamics (2020) 1–8.
 - [25] E. Jangam, C. S. R. Annavarapu, A stacked ensemble for the detection of COVID-19 with high recall and accuracy, Computers in Biology and Medicine 135 (2021) 104608.
 - [26] Z. Wang, Q. Liu, Q. Dou, Contrastive cross-site learning with redesigned net for COVID-19 CT classification, IEEE Journal of Biomedical and Health Informatics 24 (2020) 2806–2813.
 - [27] J. Zhao, Y. Zhang, X. He, P. Xie, COVID-CT-dataset: a CT scan dataset about COVID-19, arXiv preprint arXiv:2003.13865 490 (2020).
 - [28] X. Chen, L. Yao, T. Zhou, J. Dong, Y. Zhang, Momentum contrastive learning for few-shot COVID-19 diagnosis from chest CT images, Pattern recognition 113 (2021) 107826.
 - [29] 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).
 - [30] M. Polsinelli, L. Cinque, G. Placidi, A light CNN for detecting COVID-19 from CT scans of the chest, Pattern Recognition Letters 140 (2020) 95–100.
 - [31] A. Ter-Sarkisov, COVID-CT-mask-net: Prediction of COVID-19 from ct scans using regional features, medRxiv (2020).
 - [32] W. Zhao, W. Jiang, X. Qiu, Deep learning for COVID-19 detection based on CT images, Scientific Reports 11 (2021) 1–12.
 - [33] N. S. Shaik, T. K. Cherukuri, Transfer learning based novel ensemble classifier for COVID-19 detection from chest CT-scans, Computers in Biology and Medicine 141 (2022) 105127.
 - [34] R. Kundu, P. K. Singh, S. Mirjalili, R. Sarkar, COVID-19 detection from lung CT-scans using a fuzzy integral-based CNN ensemble, Computers in Biology and Medicine 138 (2021) 104895.
 - [35] 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.
 - [36] X. Chen, H. Fan, R. Girshick, K. He, Improved baselines with momentum contrastive learning, arXiv preprint arXiv:2003.04297 (2020).
 - [37] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning, PMLR, 2020, pp. 1597–1607.
 - [38] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, G. Hinton, Big self-supervised models are strong semi-supervised learners, arXiv preprint arXiv:2006.10029 (2020).
 - [39] X. Chen, K. He, Exploring simple siamese representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15750–15758.
 - [40] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, A. Joulin, Unsupervised learning of visual features by contrasting cluster assignments, arXiv preprint arXiv:2006.09882 (2020).
 - [41] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, et al., Bootstrap your own latent: A new approach to self-supervised learning, arXiv preprint arXiv:2006.07733 (2020).
 - [42] J. Li, G. Zhao, Y. Tao, P. Zhai, H. Chen, H. He, T. Cai, Multi-task contrastive learning for automatic CT and X-ray diagnosis of COVID-19, Pattern Recognition 114 (2021) 107848.
 - [43] E. D. Cubuk, B. Zoph, J. Shlens, Q. V. Le, RandAugment: Practical automated data augmentation with a reduced search space, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 702–703.
 - [44] Z. Zhong, L. Zheng, G. Kang, S. Li, Y. Yang, Random erasing data augmentation, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 13001–13008.
 - [45] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, Mixup: Beyond empirical risk minimization, arXiv preprint arXiv:1710.09412 (2017).
 - [46] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, Y. Yoo, CutMix: Regularization strategy to train strong classifiers with localizable features, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6023–6032.
 - [47] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jégou, Training data-efficient image transformers & distillation through attention, in: International Conference on Machine Learning, PMLR, 2021, pp. 10347–10357.
 - [48] R. Wightman, Pytorch image models, <https://github.com/rwightman/pytorch-image-models>, 2019. doi:10.5281/zenodo.4414861.
 - [49] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, Supervised contrastive learning, arXiv preprint arXiv:2004.11362 (2020).
 - [50] I. Loshchilov, F. Hutter, SGDR: Stochastic gradient descent with warm restarts, arXiv preprint arXiv:1608.03983 (2016).
 - [51] Y. M. Asano, C. Rupprecht, A. Vedaldi, A critical analysis of self-supervision, or what we can learn from a single image, arXiv preprint arXiv:1904.13132 (2019).
 - [52] 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.
 - [53] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, A. Vedaldi, Fine-grained visual classification of aircraft, arXiv preprint arXiv:1306.5151 (2013).
 - [54] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images (2009).
 - [55] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3606–3613.
 - [56] M.-E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, IEEE, 2008, pp. 722–729.
 - [57] O. M. Parkhi, A. Vedaldi, A. Zisserman, C. Jawahar, Cats and dogs, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 3498–3505.
 - [58] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3D object representations for fine-grained categorization, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2013, pp. 554–561.

A. Augmentation Visualization

The augmented CT scans under designed augmentation level from 1 to 6 are visualized in Fig. 4. These augmentation operations keep the main pneumonic pathologies reserved while enhancing the dataset diversity.

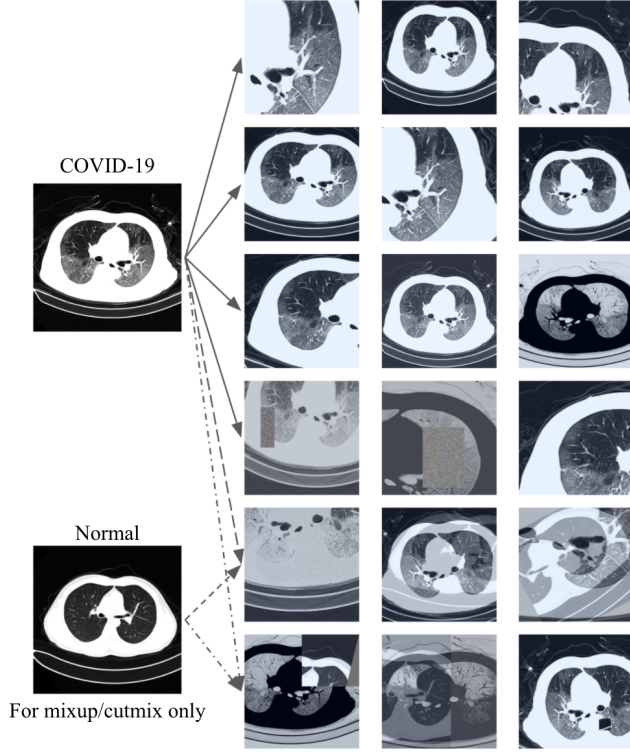


Figure 4: Illustration of applied incremental augmentation of six levels. On the right side, six groups of images are augmented from the same left COVID-19 positive scan. These groups from top to down are in the augmentation levels from 1 to 6. The bottom left normal scan is the auxiliary original image that only participated in Mixup/Cutmix augmentation in levels 5 and 6. The left two scans are from COVIDx CT-2A.

B. Effect of Projection Size

The output dimension, or named projection size, of both the projector and predictor in SR calculation are set to be 128 as default. We keep the hidden dimension 512 unchanged to avoid redundant computation while varying the projection size to analyze its effect in terms of classification accuracy. As visualized in Fig. 5, the differences between the classification accuracy for models except for SqueezeNet are small ($\leq 0.2\%$). This indicates that the hyper-parameter setting in our proposed SR is robust.

C. Attention Visualization

To understand the behavior of our model, we visualize the attention of DenseNet121-SR on three CT scans in different classes as in Fig. 6. It can be observed that

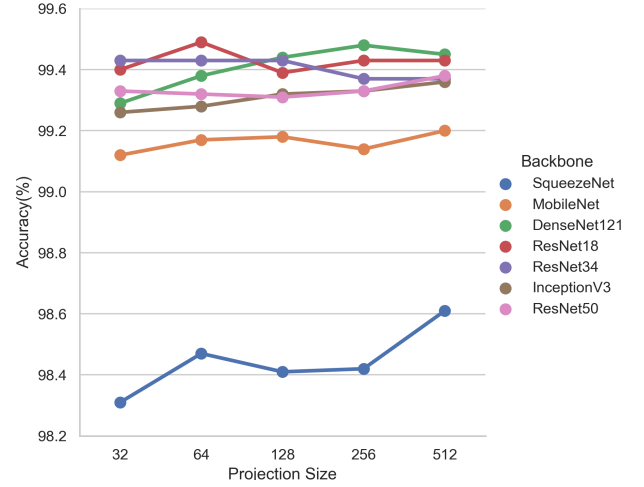


Figure 5: Impact of projection size in SR calculation.

our model mainly focuses its attention on some suspicious regions where the pathologies may exist.

D. Effect of Constant γ Strategy

The constant γ value still requires studies for finding its effects on model performance. We thus vary the γ value in constant strategy from 0.1 to 0.9 with 0.2 interval and repeat the experiments for CNNs with SR under augmentation level 4. As shown in Fig. 7, SR can improve the CNN classification performance when γ value is in an appropriate range near $[0.5, 0.7]$. In particular, a smaller γ cannot fully fulfill the advantage of SR and sometimes even degrades the model capacity as in SqueezeNet1.1 case. Meanwhile, setting γ to a large value like 0.9 is also risky since SR dominates the total loss while the primary cross entropy for classification is slighted.

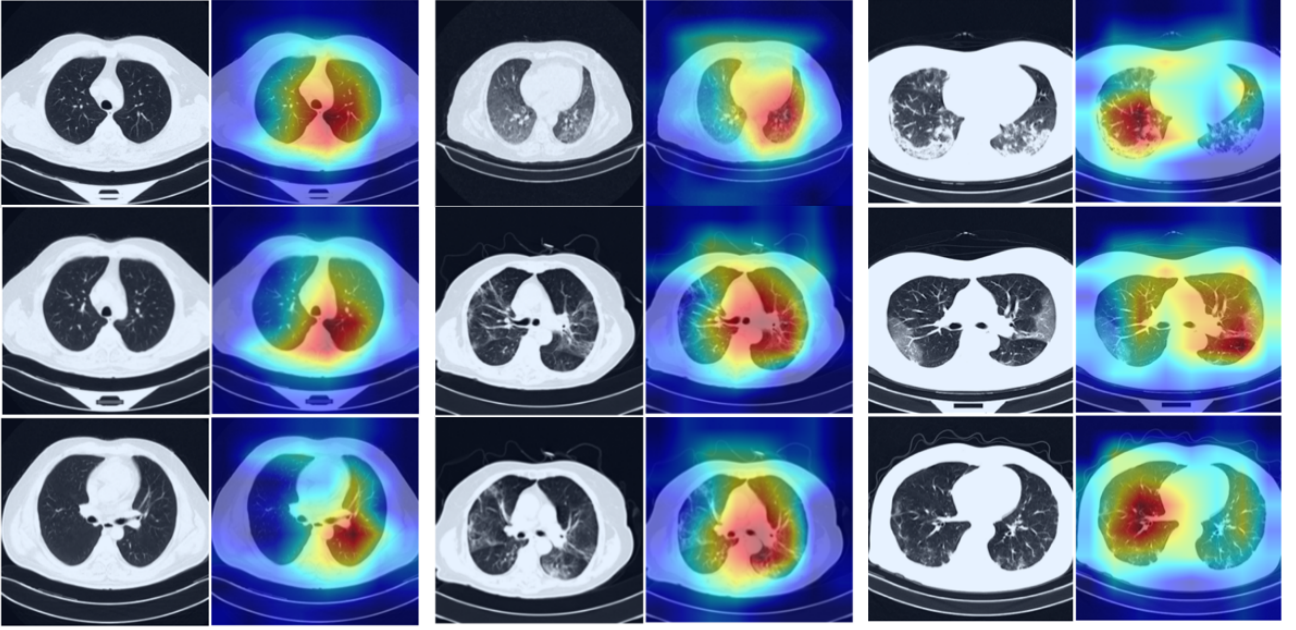


Figure 6: Attention of DenseNet121-SR visualized by Grad-CAM. The three groups of CT scans and heatmaps from left to right are in class normal, NCP, and CP, respectively. The highlighted parts are the regions based on which CNNs classify the CT scans.

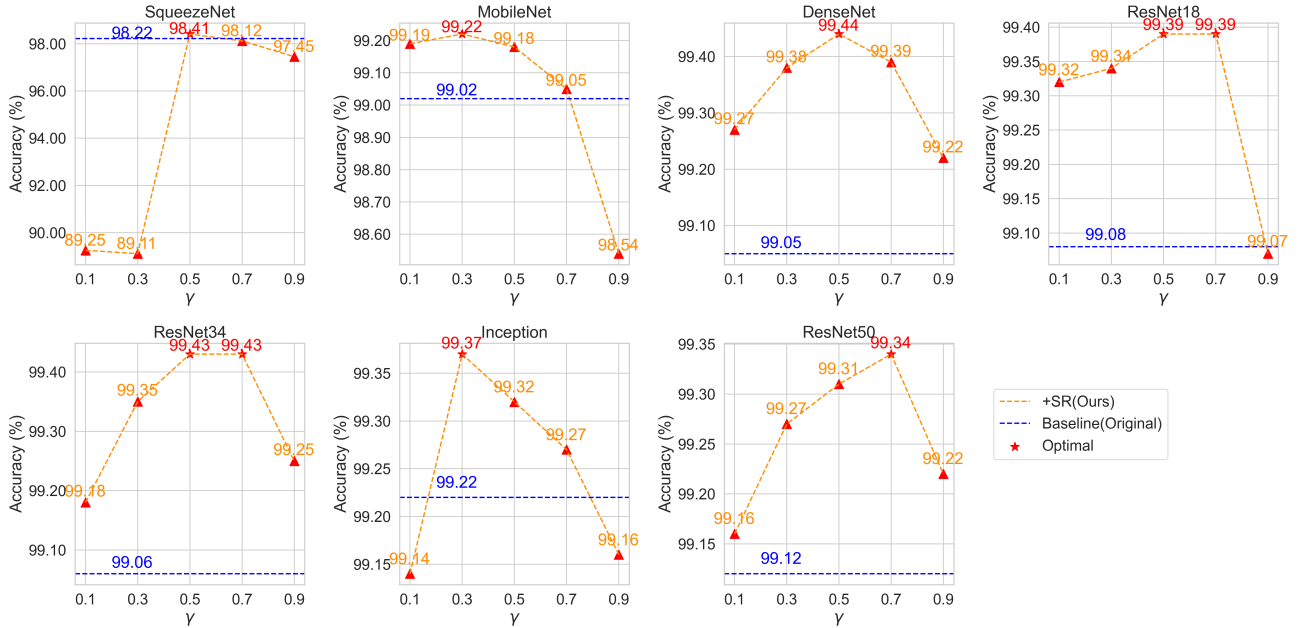


Figure 7: Accuracy of seven CNNs with SR controlled by constant γ scale factor strategy, under augmentation level 4. γ varies from 0.1 to 0.9 with 0.2 interval.