Computers, Materials & Continua
DOI:10.32604/cmc.2022.020140
Article

Tech Science Press

Deep Rank-Based Average Pooling Network for Covid-19 Recognition

Shui-Hua Wang[1], Muhammad Attique Khan[2], Vishnuvarthanan Govindaraj[3], Steven L. Fernandes[4], Ziquan Zhu[5] and Yu-Dong Zhang[6,*]

[1]School of Mathematics and Actuarial Science, University of Leicester, LE1 7RH, UK
[2]Department of Computer Science, HITEC University Taxila, Taxila, Pakistan
[3]Department of Biomedical Engineering, Kalasalingam Academy of Research and Education, 626 126, Tamil Nadu, India
[4]Department of Computer Science, Design & Journalism, Creighton University, Omaha, Nebraska, USA
[5]Science in Civil Engineering, University of Florida, Gainesville, Florida, FL 32608, USA
[6]School of Informatics, University of Leicester, UK
[*]Corresponding Author: Yu-Dong Zhang. Email: yudongzhang@ieee.org
Received: 11 May 2021; Accepted: 18 June 2021

Abstract: (Aim) To make a more accurate and precise COVID-19 diagnosis system, this study proposed a novel deep rank-based average pooling network (DRAPNet) model, i.e., deep rank-based average pooling network, for COVID-19 recognition. (Methods) 521 subjects yield 1164 slice images via the slice level selection method. All the 1164 slice images comprise four categories: COVID-19 positive; community-acquired pneumonia; second pulmonary tuberculosis; and healthy control. Our method firstly introduced an improved multiple-way data augmentation. Secondly, an *n*-conv rank-based average pooling module (NRAPM) was proposed in which rank-based pooling—particularly, rank-based average pooling (RAP)—was employed to avoid overfitting. Third, a novel DRAPNet was proposed based on NRAPM and inspired by the VGG network. Grad-CAM was used to generate heatmaps and gave our AI model an explainable analysis. (Results) Our DRAPNet achieved a micro-averaged F1 score of 95.49% by 10 runs over the test set. The sensitivities of the four classes were 95.44%, 96.07%, 94.41%, and 96.07%, respectively. The precisions of four classes were 96.45%, 95.22%, 95.05%, and 95.28%, respectively. The F1 scores of the four classes were 95.94%, 95.64%, 94.73%, and 95.67%, respectively. Besides, the confusion matrix was given. (Conclusions) The DRAPNet is effective in diagnosing COVID-19 and other chest infectious diseases. The RAP gives better results than four other methods: strided convolution, l_2-norm pooling, average pooling, and max pooling.

Keywords: COVID-19; rank-based average pooling; deep learning; deep neural network

1 Introduction

COVID-19 has caused over 158.3 million confirmed cases, with over 3.29 million death tolls till 9/May/2021. The key symptoms of COVID-19 are high temperature, new and continuous

cough, and loss or change to smell or taste [1]. Most people have mild symptoms, but some people may develop acute respiratory distress syndrome, which may trigger multi-organ failure, blood clots, septic shock, and cytokine storm.

The real-time reverse-transcriptase polymerase chain reaction technique is the main viral testing method. It usually picks the nasopharyngeal swab trials to test the presence of RNA pieces of the virus. Nevertheless, the swab could be easily contaminated, and it takes hours to two days to wait for the results [2]. Hence, chest imaging is used as an alternative way to diagnose COVID-19. Chest computed tomography (CCT) is the dominant chest imaging method compared to chest radiography and chest ultrasound since CCT can provide 3D chest imaging scans with the finest resolutions. Particularly, Ai et al. [3] carried out a large study comparing CCT against rRT-PCR, and found CCT is faster and more sensitive.

The lesions of COVID-19 in CCT are shown with main symptoms of regions of ground-glass opacity (GGO). The manual recognition works by radiologists are labor-intensive, and tedious. The manual labelling is probable to be influenced by many factors (emotion, fatigue, lethargy, etc.). In contrast, machine learning (ML) always strictly follows the instruction designed more quickly and more reliably than humans. Furthermore, the lesions of early-phase of COVID-19 patients are small and trivial, like to the nearby healthy tissues, that can be easily detected by ML algorithms meanwhile probably ignored by human radiologists.

There have been many ML methods proposed this year to recognize COVID-19 or other related diseases. Roughly speaking, those methods can be divided into traditional ML methods [4,5] and deep learning (DL) methods [6–10]. However, the performance of all those methods can still be improved. Hence, this study presents a novel DL approach: rank-based average pooling neural network with PatchShuffle (RAPNNSP). The contributions of this study entail the following four points:

(i) An improved 18-way data augmentation technique is introduced to aid the model from overfitting.
(ii) An "*n*-conv rank-based average pooling module (NRAPM)" is presented.
(iii) A new "Deep RAP Network (DRAPNet)" is proposed inspired by VGG-16 and NRAPM.
(iv) Grad-CAM is utilized to prove the explainable heat map that links with COVID-19 lesions.

2 Background on COVID-19 Detection Methods

In this Section, we briefly discuss the recent ML methods for detecting COVID-19 and other diseases. Those methods will be used as a comparison baseline in our experiment. Wu [4] used wavelet Renyi Entropy (WRE) as the feature extraction; and presented a new "three-Segment Biogeography-Based Optimization" as the classifier. Li et al. [5] used wavelet packet Tsallis entropy as a feature descriptor. The authors based on biogeography-based optimization (BO), presented a real-coded BO (RCBO) as the classifier.

The pipeline of traditional ML methods [4,5] could be categorized into two stages: feature extraction and classification. Those methods show good results in detecting COVID-19. Traditional ML methods suffer from two points: (i) a long time of feature engineering; and (ii) low performance. To solve the above two issues, modern deep neural networks, e.g., convolutional neural networks (CNNs), have been investigated and applied to COVID-19.

For instance, Cohen et al. [6] presented a COVID severity score network (CSSNet). The experiments show that the mean absolute error (MAE) is 0.78 on lung opacity score, and MAE

is 1.14 on geographic extent score. Afterward, Li et al. [7] presented a fully automatic model to recognize COVID-19 via CCT. This model is dubbed COVNet. Zhang [8] presented a 7-layer convolutional neural network for COVID-19 diagnosis (CCD). The performance yielded an accuracy of 94.03 ± 0.80 for COVID-19 against healthy people. Ko et al. [9] presented a fast-track COVID-19 classification framework (FCONet in short). Wang et al. [10] proposed DeCovNet, which is a 3D deep CNN to detect COVID-19. When using a probability threshold of 0.5, DeCovNet attained a 0.901 accuracy. Erok et al. [11] presented the imaging features of the early phase of COVID-19.

The above DL methods yield promising results in recognizing COVID-19. In order to get better results, we study the structures of those neural networks, and present a novel DRAP-Net approach, by using the mechanisms of four cutting-edge approaches: (i) multiple-way data augmentation, (ii) VGG network, (iii) rank-based average pooling, and (iv) Grad-CAM.

3 Dataset and Preprocessing

Our retrospective study was exempted by the Institutional Review Boards of local hospitals. The details of the dataset were described in Ref. [12]. 521 subjects yielded 1164 slice images via the slice level selection (SLS) method. Four types of CCT were included in the dataset: (a) COVID-19 positive; (b) community-acquired pneumonia (CAP); (c) second pulmonary tuberculosis (SPT); (d) healthy control (HC).

SLS chooses $m = \{1, 2, 3, 4\}$ slices for each subject. The average number of selected slices (ANSS, denoted by a variable M_A) per class is defined as $M_A(D_k) = \frac{M_S(D_k)}{M_P(D_k)}, k = 1, \cdots, 4$, where D_k is the category, M_S and M_P stand for the number of slices by SLS, and the number of patients, respectively. The entire ANSS is defined as $M_A = \sum_{k=1}^{4} M_S(D_k) / \sum_{k=1}^{4} M_P(D_k)$. Tab. 1 shows the demographics of the four-class subject cohort; and their corresponding triplets $[M_P, M_S, M_A]$, where M_A of the entire set is 2.23.

Table 1: Subjects and images of four classes

Class index	Class name	M_P	M_S	M_A	Class index	Class name	M_P	M_S	M_A
D_1	COVID-19	125	284	2.27	D_4	HC	139	306	2.20
D_2	CAP	123	281	2.28	Total		521	1164	2.23
D_3	SPT	134	293	2.18					

Three skilled radiologists (2 juniors: A_1 and A_2, 1 senior: B_1) are called together to curate all the images. Suppose a means one CCT slice image, g stands for the labelling. The last labelling g^F of the CCT scan a is defined as:

$$g^F[a] = \begin{cases} g[A_1, a] & g[A_1, a] == g[A_2, a] \\ f_{MV}\{g^{all}[a]\} & \text{otherwise} \end{cases}, \tag{1}$$

where f_{MV} stands for majority voting, g^{all} the labelling concatenation of all 3 radiologists (A_1, A_2, B_1), viz., $g^{all}[a] = [g(A_1, a), g(A_2, a), g(B_1, a)]$.

Define the dataset is T with five stages: the raw dataset T_R, the final preprocessed output T_P, and three temporary output T_1, T_2, T_3. The flowchart of preprocessing is displayed in Fig. 1. Let $|T|$ denotes the number of images in the dataset, which keeps the same for all five stages.

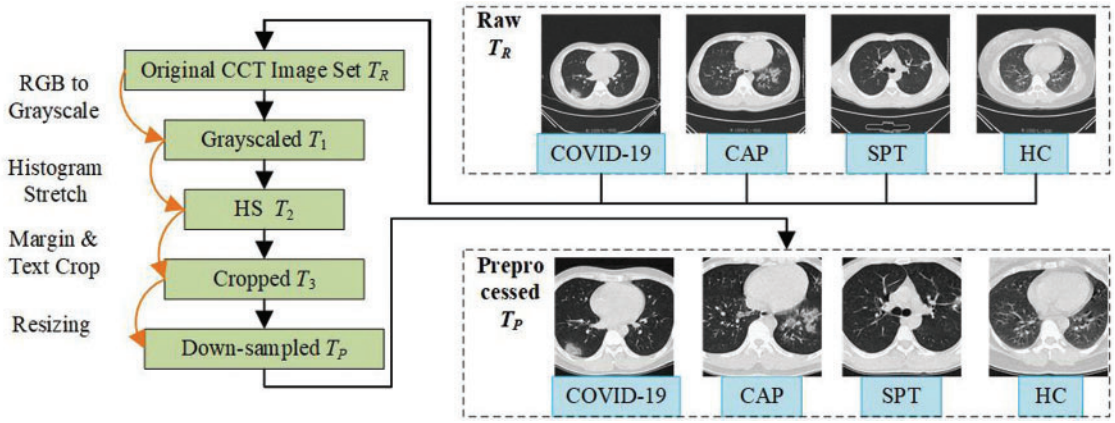

Figure 1: Illustration of preprocessing

The original raw dataset contained $|T|$ slice images $T_R = \{t_r(i), i = 1, 2, \cdots, |T|\}$. The size of each image is $size[t_r(i)] = 1024 \times 1024 \times 3$. The colorful CCT images from four classes (D_1, D_2, D_3, D_4) are transformed into grayscale versions by reserving the luminance channel. We yield the grayscale image set $T_1 = f_{Gray}(T_R)$, in which f_{Gray} stands for the grayscale operation. Note T_R are stored in three color channels, so the gray-scaling is necessary to reduce the storage.

Second, the histogram stretching (HS) is introduced for contrast-enhancement of all $|T|$ images. Select the i-th image $t_1(i), i = 1, 2, \cdots, |T|$ as an example, the minimum and maximum grayscale values $t_1^l(i)$ and $t_1^h(i)$ are reckoned as:

$$\begin{cases} t_1^l(i) = \min_{w=1}^{W_1} \min_{h=1}^{H_1} t_1(i \mid w, h), \\ t_1^h(i) = \max_{w=1}^{W_1} \max_{h=1}^{H_1} t_1(i \mid w, h), \end{cases} \tag{2}$$

where $(w,\ h)$ mean the indexes of width and height directions of the image $t_1(i)$, respectively. (W_1, H_1) stand for the width and height of the image set T_1. The recent histogram stretched data set T_2 is evaluated image-dependently, i.e., we calculate the minimum and maximum grayscale values for each image.

$$T_2 = f_{HS}(T_1) = \left\{ t_2(i) \overset{\text{def}}{=} \frac{t_1(i) - t_1^l(i)}{t_1^h(i) - t_1^l(i)} \right\}, \tag{3}$$

where f_{HS} means the histogram stretching operation.

Third, cropping is carried out to get rid of the checkup bed at the bottom area (See Fig. 1), and to remove the scripts at the corner regions. The cropped dataset T_3 is defined as: $T_3 = f_{Crop}(T_2, [b_1, b_2, b_3, b_4])$, where f_{Crop} represents crop operation. Parameters (b_1, b_2, b_3, b_4) means lengths to be cropped along four ways (top, bottom, left, and right), measured by pixels. Here the parameters (b_1, b_2, b_3, b_4) are image-independent, so they apply for all images in the dataset T_2.

Fourth, each image in T_3 is downsampled to a new image with the size of $[W_P, H_P]$, yielding the final downsized data set $T_P = f_{DS}(T_3, [W_P, H_P])$, where $f_{DS}: x \mapsto y$ stands for the

downsampling process, where y represents the down-sampled image of the original image x. The summary of the preprocessing of our method is listed in Algorithm 1.

Algorithm 1: Preprocessing in Our Method

Step 1 Import raw image set T_R,
Step 2 Grayscale and obtain $T_1 = f_{Gray}(T_R)$,
Step 3 Histogram Stretching: $T_2 = f_{HS}(T_1)$,
Step 4 Crop: $T_3 = f_{Crop}(T_2, [b_1, b_2, b_3, b_4])$,
Step 5 Downsampling: $T_P = f_{DS}(T_3, [W_P, H_P])$.

4 Methodology

4.1 Enhanced Training Set by 18-way Data Augmentation

The preprocessed dataset T_P is split into two parts: non-test set (80%) and test set (20%). Ten-fold cross-validation is performed on the non-test set to choose the optimal hyperparameter (including network structure). Afterward, 10 runs on the test set are carried out to report the test performance.

Data augmentation (DA) is an important tool to avoid overfitting and overcome the small-size dataset problem. DA has been proven to show excellent performances in many prediction/recognition/classification tasks, such as stock market prediction, prostate segmentation, etc. Recently, Wang [13] proposed a novel multiple-way data augmentation (MDA). In their 14-way DA [13], the inventors utilized seven different DA methods to the original slice $t_p(i)$ and its horizontal mirrored one $t_p^M(i)$, respectively. Later, Zhu [14] presented an 18-way DA, where they added salt-and-pepper noise (SAPN) and speckle noise (SN) to the original 14-way DA. We use the latter one, 18-way DA, in this study.

Suppose N_W stands for the number of ways of DA, that is, $N_W = 18$ in this study. For a given preprocessed image $t_p(x, y), x = 1, \cdots, W_P, y = 1, \cdots, H_P$, the SAPN altered image is defined as $t_p^{SAPN}(x, y)$, we get

$$\Pr\left(t_p^{SAPN} = t_p\right) = 1 - a_D^{SAPN}, \Pr\left(t_p^{SAPN} = v_{\min}\right) = \frac{a_D^{SAPN}}{2}, \Pr\left(t_p^{SAPN} = v_{\max}\right) = \frac{a_D^{SAPN}}{2} \tag{4}$$

where a_D^{SAPN} stands for noise density, and Pr is the probability function. $v_{\min}$ and $v_{\max}$ stand for the minimum value and maximum value of the graylevel image can have, which correspond to black and white colors, respectively. The SN altered image is defined as $t_P^{SN}(x, y) = t_P(x, y) + \mathcal{N} * t_p(x, y)$, where $\mathcal{N}$ is uniformly distributed random noise. The mean and variance of $\mathcal{N}$ are set to 0 and 0.05, respectively.

Let N_I represent the number of newly generated images for each DA, we can present the 18-way DA algorithm as follows: First, nine geometric/photometric/noise-injection DA transforms are utilized on raw image $t_p(i), i = 1, \cdots, |T|$. We use $f_{(m)}^{DA}, m = 1, \ldots, \frac{N_W}{2}$ to stand for each DA operation. It is noteworthy each DA operations f_k^{DA} generates N_I fake images. Therefore, a given image $t_p(i)$ can generate nine different data set $f_{(m)}^{DA}[t_p(i)], m = 1, \cdots, \frac{N_W}{2}$.

Second, a horizontally mirrored image is generated as $t_p^M(i) = f_M[t_p(i)]$, where f_M means horizontal mirror function.

Third, all the nine DA methods are carried out on the horizontally mirrored image $t_p^M(i)$, which generates nine new datasets $f_{(m)}^{DA}[t_p^M(i)], m = 1, \cdots, \frac{N_W}{2}$.

Fourth, the original image $t_P(i)$, the horizontally mirrored image $t_p^M(i)$, all the 9-way DA results of the original image $f_{(m)}^{DA}[t_p(i)], m = 1, \cdots, \frac{N_W}{2}$, and 9-way DA results of horizontal mirrored image $f_{(m)}^{DA}[t_p^M(i)], m = 1, \cdots, \frac{N_W}{2}$, are combined using concatenation function f_{CON}. The final combined dataset is defined as $\mathcal{T}(i)$

$$t_p(i) \mapsto \mathcal{T}(i) = f_{con} \left\{ \begin{array}{cccc} t_p(i) & \underbrace{f_{(1)}^{DA}[t_p(i)]}_{N_I} & \cdots & \underbrace{f_{(N_W/2)}^{DA}[t_p(i)]}_{N_I} \\ t_p^M(i) & \underbrace{f_{(1)}^{DA}[t_p^M(i)]}_{N_I} & \cdots & \underbrace{f_{(N_W/2)}^{DA}[t_p^M(i)]}_{N_I} \end{array} \right\} \tag{5}$$

Therefore, one image $t_p(i)$ will generate $|\mathcal{T}(i)| = N_W * N_I + 2$ images (including original image). Algorithm 2 itemizes the pseudocode of 18-way DA on one image.

Algorithm 2: Pseudocode of 18-way DA on One Training Image

Input Import the preprocessed training image $t_p(i)$.
Step 1 Nine geometric or photometric or noise-injection DA transforms are utilized on raw image $t_p(i)$, obtain $f_{(m)}^{DA}[t_p(i)], m = 1, \cdots, \frac{N_W}{2}$
Step 2 A horizontal mirror image is generated as $t_p^M(i)$.
Step 3 All the nine DAs are carried out on $t_p^M(i)$, and obtain $f_{(m)}^{DA}[t_p^M(i)], m = 1, \cdots, \frac{N_W}{2}$.
Step 4 $t_p(i)$, $t_p^M(i)$, $f_{(m)}^{DA}[t_p(i)]$, and $f_{(m)}^{DA}[t_p^M(i)]$ are combined together to form a new dataset $\mathcal{T}(i)$.
Output $\mathcal{T}(i)$ is with number of images as $N_W * N_I + 2$.

Fig. 2 shows the Step 2 result of this proposed 18-way DA results, i.e., $f_{(m)}^{DA}[t_p(i)], m = 1, \ldots, \frac{N_W}{2}$. Due to the page limit, other components, viz., $t_p^M(i)$ and $f_{(m)}^{DA}[t_p^M(i)], m = 1, \cdots, \frac{N_W}{2}$ are now displayed here. The raw image is Fig. 7a.

4.2 Proposed n-Conv Rank-Based Pooling Module

In the standard CNNs, pooling is an essential module after each convolution layer to shrink the spatial sizes of feature maps (SSFMs). Recently, strided convolution (SC) is commonly used, which also reduces SSFMs. Nevertheless, SC might be thought of as a simple pooling method, which always outputs the fixed-position value in the pooling region. In this work, we use rank-based average pooling (RAP) [15] to replace traditional max pooling. Further, RAP has been reported to yield better operation than max pooling and average pooling in up-to-date studies.

Figure 2: Results of 18-way data augmentation (a) Gaussian noise (b) SAPN (c) SN (d) Horizontal shear (e) Vertical shear (f) Image rotation (g) Scaling (h) Random translation (i) Gamma correction

Suppose there is a post-convolution feature map (FM) assigned with a variable of $H = h_{ij}(i = 1, \cdots, M \times R, j = 1, \cdots, N \times R)$. The FM could dissent into $M \times N$ blocks, where the size of each block is $R \times R$. Let us aim at the block D_{mn} that means the m-th row and n-th column block. The elements in the block D_{mn} is defined as $D_{mn} = \{d(x, y), x = 1, \cdots, R, y = 1, \cdots, R\}$.

The strided convolution (SC) traverses the input FM with strides that equal to the block's size (R, R); thus, its output is always the first element in the pooling region D_{mn}. The l2-norm pooling (L2P), max pooling (MP), and average pooling (AP) engender the l2-norm value, maximum value, and average value of the block D_{mn}, respectively. Let O be the pooling output, we have:

$$\begin{cases} O_{SC}(D_{mn}) = d(1, 1), \\ O_{L2P}(D_{mn}) = \sqrt{\frac{\sum_{x=1}^{R} \sum_{y=1}^{R} d^2(x,y)}{R^2}}, \\ O_{MP}(D_{mn}) = \max_{x=1}^{R} \max_{y=1}^{R} d(x, y), \\ O_{AP}(D_{mn}) = \frac{1}{R \times R} \sum_{x=1}^{R} \sum_{y=1}^{R} d(x, y). \end{cases} \tag{6}$$

Note that the ordinary convolutional neural network (CNN) can be combined with all the above four techniques, and we can attain SC-CNN, L2P-CNN, MP-CNN, and AP-CNN, respectively. Those four methods will be utilized as comparison baselines in the experiment.

The RAP is not a value-based pooling; in contrast, SP is a type of rank-based pooling. The output of RAP is based on ranks of pixels other than values of pixels in the block D_{mn}. Thus, RAP could solve the shortcomings of MP and AP. MP outputs the maximum value but worsens the overfitting challenge. Oppositely, the AP produces the average, with the shortcoming of downscaling the largest value, where the important traits may be contained.

RAP is a three-step route. First, the ranking matrix (RM) $T = \{t_{xy}\}$ is generated from the pooling region, where $x = 1, \cdots, R$, $y = 1, \cdots, R$ and $t_{xy} \in [1, 2, \cdots, R \times R]$. In all, T is generated by the rule: the less value the entry is, the higher value the rank is. If tied values are for $d(x1, y1)$ and $d(x2, y2)$, then we check the index values of $x1$ and $x2$. If $x1$ equals $x2$, then we check the value of $y1$ and $y2$.

$$\begin{cases} d(x1, y1) < d(x2, y2) \rightarrow t_{(x1,y1)} > t_{(x2,y2)}, \\ [d(x1, y1) == d(x2, y2)] \wedge (x1 > x2) \rightarrow t_{(x1,y1)} > t_{(x2,y2)}, \\ [d(x1, y1) == d(x2, y2)] \wedge (x1 == x2) \wedge (y1 > y2) \rightarrow t_{(x1,y1)} > t_{(x2,y2)}. \end{cases} \tag{7}$$

Second, select the pixels whose ranks are no more than a threshold δ_{RAP}, which controls how many pixels within a region will be considered. The selected elements are rearranged into a candidate vector (CV) as $v_{CV} = \{d(x, y) | 1 \leq t_{xy} \leq \delta_{RAP}\}$.

Third, the average CV v_{RAP} is output as final RAP output:

$$O_{RAP}(D_{mn}) = \frac{1}{\delta_{RAP}} \sum v_{CV} = \frac{1}{\delta_{RAP}} \sum d(x, y) | 1 \leq t_{xy} \leq \delta_{RAP}. \tag{8}$$

Algorithm 3: Pseudocode of RAP

Step 1 Input: pooling region $D_{mn} = \{d_{xy}\}$,
Step 2 Generate ranking matrix: $D_{mn} \mapsto T = \{t_{xy}\}$, See Eq. (7)
Step 3 Select entries whose ranks are less than δ_{RAP}: $v_{CV} = d(x, y)|1 \leq t_{xy} \leq \delta_{RAP}$.
Step 4 Calculate the average of v_{RAP}: $O_{RAP} = \sum v_{cv}/\delta_{RAP}$.
Step 5 Output: O_{RAP}.

Algorithm 3 shows the pseudocode of RAP. Fig. 3 shows the comparison of four different pooling methods, where $\delta_{RAP} = 4$ and $R = 3$. Select the top-left block (in red rectangle) as an example, it contains 9 entries as: 2.4, 8.9, 4.9, 9.2, 9.5, 0.5, 3.4, 5.4, and 3.4. The L2P, AP, and MP output 6.88, 6.29, and 9.5 respectively, using Eq. (6). In contrast, RAP first calculate the RM and selects the δ_{RAP} greatest entries, i.e., $v_{RAP} = (9.5, 9.5, 9.2, 8.9)$. The average of v_{RAP} is the output of RAP, thus $O_{RAP} = 9.28$.

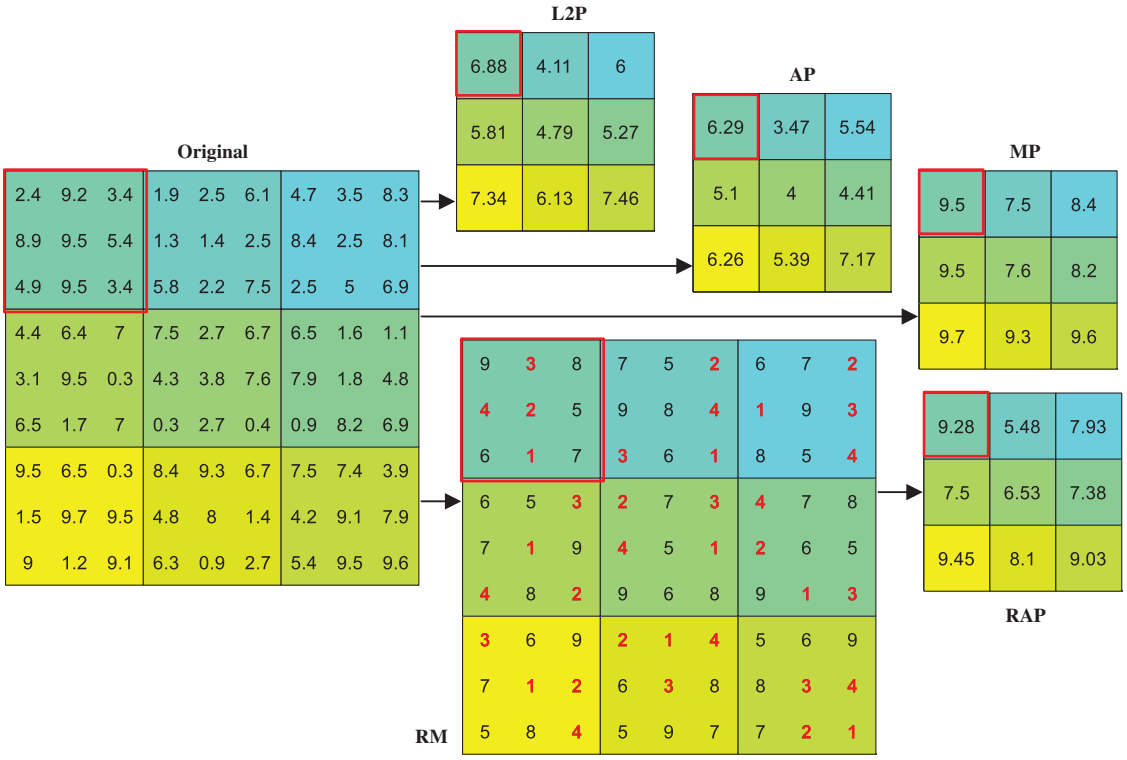

Figure 3: Comparison of four pooling methods

The second contribution of this study is that we proposed a new "n-conv rank-based average pooling module" (NRAPM) based on RAP layer. The NRAPM is composed of n-repetitions of a conv layer and a BN layer, followed by a RAP layer. Fig. 4 displays the graph of the proposed NRAPM. We set $1 \leq n \leq 3$, because we run our model using $n > 3$ on training set, but the validation performance of $n > 3$ did not improve. ReLU function is missing in Fig. 4.

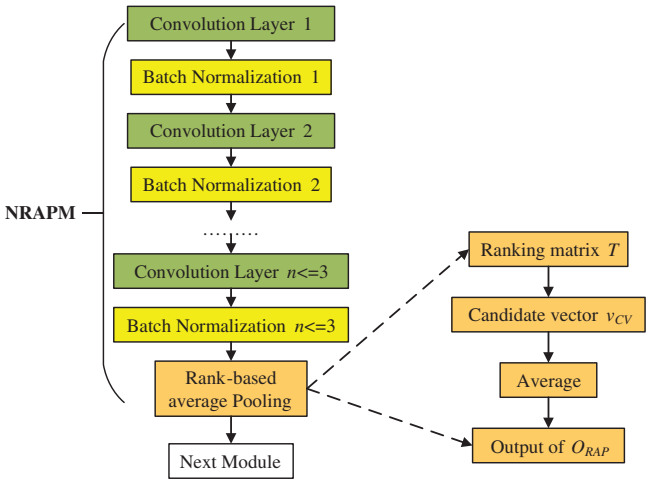

Figure 4: Schematic of proposed NRAPM

4.3 DRAPNet: Deep RAP Network

The final contribution of this study is to propose a deep RAP network (DRAPNet) with its conv block being NRAPM and its structure inspired by VGG-16 [16]. Fig. 5 displays the structure of VGG-16, which entails five conv layers and three dense layers (i.e., fully connected layer). The input of VGG-16 is $224 \times 224 \times 3$. After the 1st convolution block (CB), that entails (i) two repetitions of 2 convolutional layers with 64 kernels whose sizes are 3×3, and (ii) one max pooling layer with a size of 2×2. The 1st CB is abbreviated as "2 × (64 3 × 3)". The output of 1st CB is $112 \times 112 \times 64$.

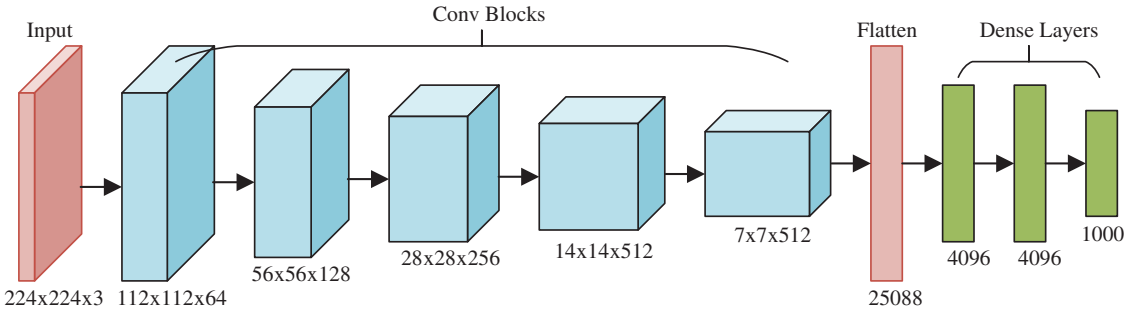

Figure 5: Structure of VGG-16

The 2nd CB is 2 × (128 3 × 3), 3rd CB 3 × (256 3 × 3), 4th CB 3 × (512 3 × 3), and 5th CB 3 × (512 3 × 3). Those generate the FMs with sizes of $56 \times 56 \times 128$, $28 \times 28 \times 256$, $14 \times 14 \times 512$, and $7 \times 7 \times 512$, respectively. Later, the FM of 5th CB is compressed into a vector of 25,088 neurons and delivered into three dense layers with the number of neurons of 4096, 4096, and 1000, respectively.

Inspired by VGG-16, this proposed DRAPNet network uses a small conv kernel other than a large kernel, and always uses 2x2 filters with a stride of 2 for pooling. Besides, both DRAPNet and VGG-16 employ repetitions of conv layers followed by pooling as a CB. They both use dense layers at the end. The structure of DRAPNet is adjusted by validation performance and itemized

in Tab. 2, in which NWL represents the number of weighted layers, CH the configuration of hyperparameters.

Table 2: Detailed structure of 12-layer DRAPNet (I = Index)

I	Layer	NWL	CH	SSFM	I	Layer	NWL	CH	SSFM
1	Input	0	0	$256 \times 256 \times 1$	6	NRAPM-5	3	$3 \times [3 \times 3, 128]$	$8 \times 8 \times 128$
2	NRAPM-1	1	$1 \times [3 \times 3, 32]$	$128 \times 128 \times 32$	7	Flatten	0	0	$1 \times 1 \times 8192$
3	NRAPM-2	2	$2 \times [3 \times 3, 32]$	$64 \times 64 \times 32$	8	FCL-1	1	$150 \times 8192, 150 \times 1$	$1 \times 1 \times 150$
4	NRAPM-3	2	$2 \times [3 \times 3, 64]$	$32 \times 32 \times 64$	9	FCL-2	1	$4 \times 150, 4 \times 1$	$1 \times 1 \times 4$
5	NRAPM-4	2	$2 \times [3 \times 3, 64]$	$16 \times 16 \times 64$					

Compared to standard CNNs, the gains of DRAPNet include two points: (i) DRAPNet facilitates our model from overfitting by using the proposed NRAPM; (ii) DRAPNet is parameter-free. (iii) DRAPNet can be straightforwardly united with other enhanced network mechanisms, e.g., dropout, etc. Overall, we build this 12-layer DRAPNet. We have endeavored to incorporate more NRAPMs or more FCLs, which do not improve the functioning but adding more calculation loads.

Take a close-up of Tab. 2, the CH column in the top part has a format of $a \times [b \times b, c]$, which stands of a repetitions of c filters with size of $b \times b$. For the bottom part of Tab. 2, the $d \times e, f \times g$ format in CH column means the weight matrix with size of $d \times e$ and bias vector with size of $f \times g$. In the SSFM column of Tab. 2, the format of $h \times i \times j$ stands for the spatial size of feature maps, where h, i, j represents height, width, and channel, respectively.

Tab. 3 itemizes the non-test, and test set for each class. The whole dataset T_P comprises four non-overlapping categories $T_P = \{T_P^1, T_P^2, T_P^3, T_P^4\}$. For each class, the dataset is split into non-test set and test set $T_P^k \to \{A_P^k, B_P^k\}, k = 1, 2, 3, 4$, where A_P, B_P mean the preprocessed non-test set, and preprocessed test set respectively

$$T_P \overset{\text{def}}{=} \begin{bmatrix} T_P^1 \\ T_P^2 \\ T_P^3 \\ T_P^4 \end{bmatrix} \overset{\text{def}}{=} \begin{bmatrix} \underbrace{A_P}_{\text{non-test}} & \underbrace{B_P}_{\text{test}} \end{bmatrix} \overset{\text{def}}{=} \begin{bmatrix} A_P^1 & B_P^1 \\ A_P^2 & B_P^2 \\ A_P^3 & B_P^3 \\ A_P^4 & B_P^4 \end{bmatrix}. \tag{9}$$

The experiment involves two stages. At Stage I, 10-fold cross-validation is utilized on the non-test set A_P, to fix the best network structure and hyperparameters. The 18-way DA is utilized on the training set during 10-fold cross-validation.

Afterward, at Stage II, this DRAPNet model is trained using a non-test set A_P as training set, and evaluated using a test set B_P as test set. Again, 18-way DA is used on the training set. The algorithm run B_R times with different initial seeds. Once combining the B_R runs, we attain a summation of a test confusion matrix B_M. Tab. 3 itemizes how the dataset is split, where $|a|$ means the number of elements of the set a.

Table 3: Dataset splitting

Category	Non-test (10-fold cross validation)	Test (10 runs)	Total
COVID-19	$\lvert A_P^1\rvert = 227$	$\lvert B_P^1\rvert = 57$	$\lvert T_P^1\rvert = 284$
CAP	$\lvert A_P^2\rvert = 225$	$\lvert B_P^2\rvert = 56$	$\lvert T_P^2\rvert = 281$
SPT	$\lvert A_P^3\rvert = 234$	$\lvert B_P^3\rvert = 59$	$\lvert T_P^3\rvert = 293$
HC	$\lvert A_P^4\rvert = 245$	$\lvert B_P^4\rvert = 61$	$\lvert T_P^4\rvert = 306$

The ideal $B_M = \{b_m(i,j), i = 1,\ldots,4, j = 1,\ldots,4\}$ is a diagonal matrix with the appearance of

$$B_M^{ideal} = \{b_m^{ideal}(i,j)\} = B_R \times \begin{bmatrix} |B_P^1| & 0 & 0 & 0 \\ 0 & |B_P^2| & 0 & 0 \\ 0 & 0 & |B_P^3| & 0 \\ 0 & 0 & 0 & |B_P^4| \end{bmatrix}, \tag{10}$$

where all the off-diagonal elements' values are zero, i.e., $b_m^{ideal}(i,j) = 0, \forall i \neq j$, indicating no classification errors. In realistic AI models that make errors, the performance indicators are computed per class. For each class $k = 1, 2, 3, 4$, hat class label k is set as "positive", and all other three classes $f_{SD}[(1,2,3,4),k]$ are "negative", where f_{SD} means the set difference function. Three performance indicators (sensitivity, precision, and F1 score) of class k are defined as:

$$Sen(k) = TP(k)/[TP(k) + FN(k)]. \tag{11}$$

$$Prc(k) = TP(k)/[TP(k) + FP(k)]. \tag{12}$$

$$F1(k) = [2 \times Prc(k) \times Sen(k)]/[Prc(k) + Sen(k)]. \tag{13}$$

The test performance could be calculated over all four classes. The micro-averaged (MA) F1 (denoted as F_m) is harnessed, due to the slightly unbalance of our dataset

$$F_m = \frac{2 \times \frac{\sum_{k=1}^4 TP(k)}{\sum_{k=1}^4 TP(k)+FP(k)} \times \frac{\sum_{k=1}^4 TP(k)}{\sum_{k=1}^4 TP(k)+FN(k)}}{\frac{\sum_{k=1}^4 TP(k)}{\sum_{k=1}^4 TP(k)+FP(k)} + \frac{\sum_{k=1}^4 TP(k)}{\sum_{k=1}^4 TP(k)+FN(k)}}. \tag{14}$$

Lastly, gradient-weighted class activation mapping (Grad-CAM) is used to give clarifications on how this DRAPNet model renders the decision and which region it pays more attention to. Grad-CAM employs the gradient of the categorization score with regards to the convolutional features decided by our model. The FM of NRAPM-5 in Tab. 2 is harnessed for Grad-CAM.

5 Experiments, Results, and Discussions

Some common parameters are itemized in Tab. 4. The crop parameters are set as $b_1 = b_2 = b_3 = b_4 = 200$. The size of final preprocessed image is $W_P = H_P = 256$. The noisy density of SAPN is 0.05. The number of newly generated images of each DA is set as $N_I = 30$. We tested greater value of N_I, however, it does not yield substantial advances on the validation set. The number of ways of DA is set to $N_W = 18$. For RAP, only 2 elements will be selected for each pooling region.

The number of runs on the test set is set to $B_R = 10$. Besides, the operating system is Windows 10. The programming environment is MATLAB 2021a. GPU device is NVIDIA GeForce GTX1060.

Table 4: Parameter setting

Parameter	Value	Parameter	Value	Parameter	Value
b_1, b_2, b_3, b_4	200	a_D^{SAPN}	0.05	δ_{RAP}	2
W_P	256	N_I	30	B_R	10
H_P	256	N_W	18		

5.1 Confusion Matrix of Proposed DRAPNet Model

Fig. 6 shows the confusion matrix of DRAPNet with 10 runs over the test set. Each row represents the number of samples in the true class, and each column represents the number of samples in the predicted class. The entry $a(i,j)$ in this confusion matrix A stands for the number of cases in class i predicted as class j. Blue color (diagonal entries) and pink color (off-diagonal entries) represent the correct and incorrect observations, respectively.

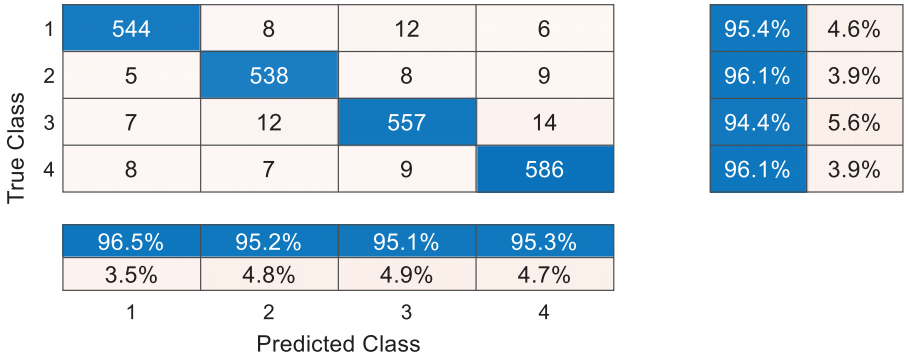

Figure 6: Confusion matrix with 10 runs over test set

Take a close-up to Fig. 6, the sensitivities of four classes are 95.44%, 96.07%, 94.41%, and 96.07%, respectively. The precisions of four classes are 96.45%, 95.22%, 95.05%, and 95.28%, respectively. The F1 scores of the four classes are not shown in Fig. 6, and their values are 95.94%, 95.64%, 94.73%, and 95.67%, respectively. The micro-averaged F1 is 95.49%.

5.2 Comparison of DRAPNet and Other Pooling Methods

Proposed DRAPNet is compared against the other four CNNs with various pooling techniques. Those five CNNs are SC-CNN, L2P-CNN, MP-CNN, and AP-CNN, respectively. Their description can be found in Section 4.2. Take SC-CNN as an example, it uses the same structure of DRAPNet but replaces RAP with SC. The results of 10 runs of those five methods over the test set are displayed in Tab. 5, where C represents class, (D_1, D_2, D_3, D_4) stands for the four classes.

Table 5: Comparison of DRAPNet with four standard CNNs

Model	C	Sen	Prc	F1	Model	C	Sen	Prc	F1	Model	C	Sen	Prc	F1
SC-CNN	D_1	95.09	93.77	94.43	L2P-CNN	D_1	92.28	94.77	93.51	AP-CNN	D_1	93.33	94.66	93.99
	D_2	93.21	93.88	93.55		D_2	95.54	93.53	94.52		D_2	93.21	91.58	92.39
	D_3	92.03	94.43	93.22		D_3	91.69	90.77	91.23		D_3	94.58	93.78	94.18
	D_4	93.11	91.47	92.28		D_4	93.44	93.90	93.67		D_4	95.08	96.19	95.63
	MA			93.35		MA			93.22		MA			94.08
MP- CNN	D_1	91.75	94.40	90.87	DRAPNet (Ours)	D_1	95.44	96.45	95.94					
	D_2	94.64	93.47	92.16		D_2	96.07	95.22	95.64					
	D_3	95.42	93.83	92.78		D_3	94.41	95.05	94.73					
	D_4	95.90	96.06	94.56		D_4	96.07	95.28	95.67					
	MA			92.62		MA			95.49					

There are in total 13 indicators, and we choose to use micro-averaged F1 as the main indicator since it takes the performances of all categories into consideration. The micro-averaged F1 scores of SC-CNN, L2P-CNN, MP-CNN, AP-CNN, and DRAPNet are 93.35%, 93.22%, 92.62%, 94.08%, and 95.49%, respectively. The reason why DRAPNet obtains the best micro-averaged F1 score is that RAP can prevent overfitting [15], which is the main shortcoming of max pooling. Meanwhile, L2P and AP average out the maximum activation values, that hurdle the performance of the corresponding L2P-CNN and AP-CNN models. For SC-CNN, it barely employs one-fourth of all knowledge of the input FM; and neglects the rest three-fourths of information; thus, its performance is not comparable to RAP.

5.3 Comparison to State-of-the-Art Approaches

This proposed DRAPNet method is compared with 8 state-of-the-art methods: WRE [4], RCBO [5], CSSNet [6], COVNet [7], CCD [8], FCONet [9], DeCovNet [10], VGG-16 [16]. All the experiments are implemented on the same test set by 10 runs. Comparison results are itemized in Tab. 6, where C represents class, (D_1, D_2, D_3, D_4) stands for the four classes. It is observed that the DRAPNet yields the greatest performance in terms of MA F1 and most of other indicators. The reason why this proposed DRAPNet performs the best is four reasons: (i) We use 18-way data augmentation to avoid overfitting, (ii) Our network is inspired by VGG, (iii) rank-based average pooling is used to replace traditional pooling, and (iv) Grad-CAM is used to provide explainability of our DRAPNet model.

5.4 Explainability

We take four samples (one sample per category) as examples, the raw images of those four pictures are shown in Figs. 7a–7d, their corresponding heatmaps are shown in Figs. 7e–7h, and the cognate manual delineation results are shown in Figs. 7i–7k. It is noteworthy there are no lesions within healthy subject images.

The FM of NRAPM-5 in DRAPNet is used to generate the heat maps by Grad-CAM. We can see from Fig. 7 that the heatmaps by this DRAPNet model and Grad-CAM are able to apprehend the diseased lesions efficiently and to ignore those non-lesion areas. Conventionally, AI is viewed as a "black box", which hurdles its widespread use. Nevertheless, with the help of explainability of modern AI techniques, the radiologist and patients could gain assurances to this DRAPNet model, since the heat map gives a self-explanatory interpretation of how AI classifies COVID-19, CAP, SPT from healthy subjects.

Table 6: Comparison with state-of-the-art approaches

Model	C	Sen	Prc	F1	Model	C	Sen	Prc	F1	Model	C	Sen	Prc	F1
WRE [4]	D_1	82.63	83.96	83.29	RCBO [5]	D_1	71.93	84.19	77.58	CSS Net [6]	D_1	94.04	92.25	93.14
	D_2	80.89	76.26	78.51		D_2	72.86	72.73	72.79		D_2	93.75	95.11	94.42
	D_3	84.58	85.15	84.86		D_3	73.56	76.41	74.96		D_3	91.36	93.58	92.45
	D_4	81.15	84.04	82.57		D_4	80.66	68.91	74.32		D_4	94.43	92.75	93.58
	MA			82.32		MA			74.85		MA			93.39
COV Net [7]	D_1	89.82	86.63	88.20	CCD [8]	D_1	89.47	93.58	91.48	FCO Net [9]	D_1	92.28	95.64	93.93
	D_2	89.82	92.63	91.21		D_2	93.93	92.44	93.18		D_2	96.79	94.43	95.59
	D_3	93.73	90.66	92.17		D_3	93.73	95.18	94.45		D_3	94.75	95.88	95.31
	D_4	87.38	90.96	89.13		D_4	95.08	91.34	93.17		D_4	94.92	92.94	93.92
	MA			90.17		MA			93.09		MA			94.68
DeCov Net [10]	D_1	91.05	90.58	90.81	VGG-16 [16]	D_1	78.07	77.93	78.00	DRAP Net (Ours)	D_1	95.44	96.45	95.94
	D_2	93.75	90.99	92.35		D_2	83.21	82.48	82.84		D_2	96.07	95.22	95.64
	D_3	90.51	86.97	88.70		D_3	82.54	77.30	79.84		D_3	94.41	95.05	94.73
	D_4	88.69	95.58	92.01		D_4	70.00	75.71	72.74		D_4	96.07	95.28	95.67
	MA			90.94		MA			78.33		MA			95.49

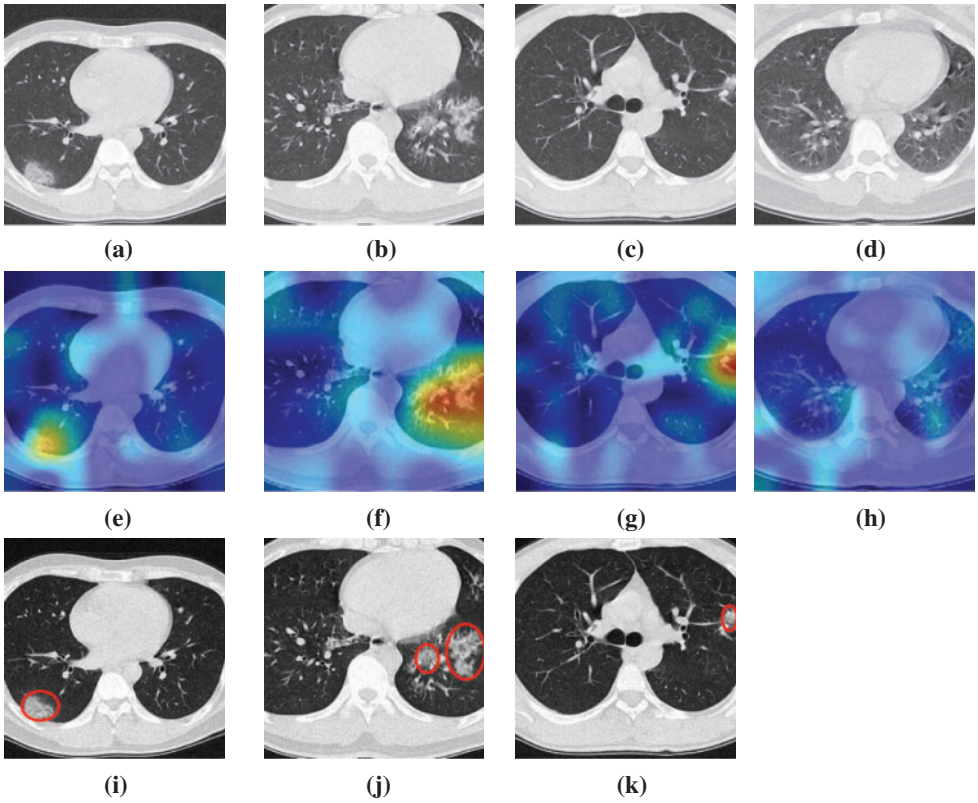

Figure 7: Heatmaps of three diseased samples and one healthy sample (a) A sample of COVID-19 (b) A sample of CAP (c) A sample of SPT (d) A sample of HC (e) Heatmap of COVID-19 (f) Heatmap of CAP (g) Heatmap of SPT (h) Heatmap of HC (i) Lesion of (a) (j) Lesion of (b) (k) Lesion of (c)

6 Conclusion

This study proposes a DRAPNet that fuses four improvements: (a) proposed NRAPM module, (b) usage of rank-based average pooling; (c) multiple-way DA; and (d) explainability via Grad-CAM. These four improvements make our DRAPNet method yield better results than 8 state-of-the-art methods. The 10 runs on the test set demonstrate this DRAPNet model achieved a micro-averaged F1 score of 95.49%.

There are three aspects that can be improved in future studies: (a) Our DRAPNet method does not go through stringent clinical validation, so we will try to develop web apps based on the mode, and deploy our apps online, and invite radiologists and physicians to return feedbacks so we can continually improve it; (b) Data collection is still ongoing, and we expect to collect more images; (c) Segmentation techniques can be used within the preprocessing to remove unrelated regions prior to the DRAPNet model.

Funding Statement: This study is partially supported by the Medical Research Council Confidence in Concept Award, UK (MC_PC_17171); Royal Society International Exchanges Cost Share Award, UK (RP202G0230); Hope Foundation for Cancer Research, UK (RM60G0680); British Heart Foundation Accelerator Award, UK; Sino-UK Industrial Fund, UK (RP202G0289); Global Challenges Research Fund (GCRF), UK (P202PF11). We thank Dr. Hemil Patel for his help in English correction.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Bodecka, I. Nowakowska, A. Zajenkowska, J. Rajchert, I. Kazmierczak *et al.,* "Gender as a moderator between present-hedonistic time perspective and depressive symptoms or stress during covid-19 lock-down," *Personality and Individual Differences*, vol. 168, pp. 7, Article ID: 110395, 2021.

[2] C. Younes, "Fecal calprotectin and rt-pcr from both nasopharyngeal swab and stool samples prior to treatment decision in ibd patients during covid-19 outbreak," *Digestive and Liver Disease*, vol. 52, pp. 1230–1230, 2020.

[3] T. Ai, Z. Yang, H. Hou, C. Zhan, C. Chen *et al.,* "Correlation of chest ct and rt-pcr testing in coronavirus disease 2019 (covid-19) in China: A report of 1014 cases," *Radiology*, vol. 296, no. 2, pp. 32–40, 2020.

[4] X. Wu, "Diagnosis of covid-19 by wavelet renyi entropy and three-segment biogeography-based optimization," *International Journal of Computational Intelligence Systems*, vol. 13, pp. 1332–1344, 2020.

[5] P. Li and G. Liu, "Pathological brain detection via wavelet packet tsallis entropy and real-coded biogeography-based optimization," *Fundamenta Informaticae*, vol. 151, pp. 275–291, 2017.

[6] J. P. Cohen, L. Dao, P. Morrison, K. Roth, Y. Bengio *et al.,* "Predicting covid-19 pneumonia severity on chest x-ray with deep learning," *Cureus*, vol. 12, Article ID: e9448, 2020.

[7] L. Li, L. Qin, Z. Xu, Y. Yin, X. Wang *et al.,* "Using artificial intelligence to detect covid-19 and community-acquired pneumonia based on pulmonary ct: Evaluation of the diagnostic accuracy," *Radiology*, vol. 296, pp. E65–E71, 2020.

[8] Y. D. Zhang, "A seven-layer convolutional neural network for chest ct based covid-19 diagnosis using stochastic pooling," *IEEE Sensors Journal*, pp. 1–1, 2020 (Online First).

[9] H. Ko, H. Chung, W. S. Kang, K. W. Kim, Y. Shin *et al.,* "Covid-19 pneumonia diagnosis using a simple 2d deep learning framework with a single chest ct image: Model development and validation," *Journal of Medical Internet Research*, vol. 22, pp. 13, Article ID: e19569, 2020.

[10] X. G. Wang, X. B. Deng, Q. Fu, Q. Zhou, J. P. Feng *et al.,* "A weakly-supervised framework for covid-19 classification and lesion localization from chest ct," *IEEE Transactions on Medical Imaging*, vol. 39, pp. 2615–2625, 2020.

[11] B. Erok and A. O. Atca, "Chest ct imaging features of early phase covid-19 pneumonia," *Acta Medica Mediterranea*, vol. 37, pp. 501–507, 2021.

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

[13] S.-H. Wang, "Covid-19 classification by fgcnet with deep feature fusion from graph convolutional network and convolutional neural network," *Information Fusion*, vol. 67, pp. 208–229, 2021.

[14] W. Zhu, "Anc: Attention network for covid-19 explainable diagnosis based on convolutional block attention module," *Computer Modeling in Engineering & Sciences*, vol. 172, no. 3, pp. 1037–1058, 2021.

[15] Z. L. Shi, Y. D. Ye and Y. P. Wu, "Rank-based pooling for deep convolutional neural networks," *Neural Networks*, vol. 83, pp. 21–31, 2016.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. on Learning Representations (ICLR)*, San Diego, CA, USA, pp. 1–14, 2015.