

## Research Article

# PSSPNN: PatchShuffle Stochastic Pooling Neural Network for an Explainable Diagnosis of COVID-19 with Multiple-Way Data Augmentation

Shui-Hua Wang<sup>1,2</sup>, Yin Zhang,<sup>3</sup> Xiaochun Cheng<sup>4</sup>, Xin Zhang<sup>5</sup>,  
and Yu-Dong Zhang<sup>6</sup>

<sup>1</sup>School of Computer Science, Henan Polytechnic University, China, Henan 454001, China

<sup>2</sup>School of Architecture Building and Civil Engineering, Loughborough University, Loughborough LE11 3TU, UK

<sup>3</sup>School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

<sup>4</sup>School of Science & Technology, Middlesex University, London NW4 4BT, UK

<sup>5</sup>Department of Medical Imaging, The Fourth People's Hospital of Huai'an, Huai'an, Jiangsu Province 223002, China

<sup>6</sup>School of Informatics, University of Leicester, Leicester LE1 7RH, UK

Correspondence should be addressed to Xiaochun Cheng; x.cheng@mdx.ac.uk, Xin Zhang; 973306782@qq.com, and Yu-Dong Zhang; yudongzhang@ieee.org

Received 9 November 2020; Revised 23 December 2020; Accepted 18 February 2021; Published 10 March 2021

Academic Editor: Martti Juhola

Copyright © 2021 Shui-Hua Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Aim.** COVID-19 has caused large death tolls all over the world. Accurate diagnosis is of significant importance for early treatment. **Methods.** In this study, we proposed a novel PSSPNN model for classification between COVID-19, secondary pulmonary tuberculosis, community-captured pneumonia, and healthy subjects. PSSPNN entails five improvements: we first proposed the n-conv stochastic pooling module. Second, a novel stochastic pooling neural network was proposed. Third, PatchShuffle was introduced as a regularization term. Fourth, an improved multiple-way data augmentation was used. Fifth, Grad-CAM was utilized to interpret our AI model. **Results.** The 10 runs with random seed on the test set showed our algorithm achieved a microaveraged F1 score of 95.79%. Moreover, our method is better than nine state-of-the-art approaches. **Conclusion.** This proposed PSSPNN will help assist radiologists to make diagnosis more quickly and accurately on COVID-19 cases.

## 1. Introduction

COVID-19 is a type of disease caused by a new strain of coronavirus. “CO” means corona, “VI” virus, and “D” disease. Up to 22 December 2020, COVID-19 has caused more than 78.0 million confirmed cases and over 1.71 million deaths (US 326.5 k, Brazil 188.2 k, India 146.1 k, Mexico 119.4 k, Italy 69.8 k, and UK 68.3 k).

To diagnose COVID-19, two methods exist: (i) real-time reverse transcriptase PCR with nasopharyngeal swab samples to test the existence of RNA fragments and (ii) chest imaging (CI) examines the evidence of COVID-19. The first type of rRT-PCR method needs to wait for a few days to get the results,

while the second type of CI approach could get quick results within minutes. The CI approaches have several advantages compared to rRT-PCR. First, the swab may be contaminated [1, 2]. Second, CI can detect the lesions of lungs where “ground-glass opacity (GGO)” will be observed to distinguish COVID-19 from healthy subjects. Third, CI can provide an immediate result as soon as imaging is complete. Fourth, reports show that chest computed tomography (CCT), one type of CI approach, can detect 97% of COVID-19 infections [3].

Currently, there are three types of CI approaches: chest X-ray (CXR) [4], chest CT (CCT), and chest ultrasound (CUS) [5]. Among all three types of approaches, CCT can provide the finest resolution than the other two CI approaches,

allowing visualization of extremely small nodules in the lung. The additional advantage of CCT is that it can provide high-quality, three-dimensional chest data where radiologists can clearly view the COVID-19 lesions, which may be obscure in the other two CI approaches.

However, manual labeling by human experts is tedious, labor-intensive, and time-consuming [6]. Besides, the labeling performances are easily affected by interexpert and intraexpert factors (e.g., emotion, tiredness, and lethargy). Moreover, the labeling throughputs of radiologists are not comparable with artificial intelligence (AI) models. For example, senior radiologists may diagnose one scanning within five minutes, but AI can analyze thousands of samples within one minute. Particularly, early lesions are small and similar to surrounding normal tissues, which make them more challenging to measure and hence can potentially be neglected by radiologists.

Traditional AI methods have successfully been applied in many medical fields. For instance, Wang et al. [7] chose wavelet packet Tsallis entropy as a feature descriptor and employed a real-coded biogeography-based optimization (RCBO) classifier. Jiang and Zhang [8] proposed a 6-level convolutional neural network (6L-CNN) for therapy and rehabilitation. Their performances were improved by replacing the traditional rectified linear unit with a leaky rectified linear unit. Fulton et al. [9] used ResNet-50 (RN-50) to classify Alzheimer’s disease with and without imagery. The authors found that ResNet-50 models help identify AD patients. Guo and Du [10] utilized a ResNet-18 (RN-18) model to recognize thyroid ultrasound standard plane (TUSP), achieving a classification accuracy of 83.88%. The experiments verified the effectiveness of RN-18. The aforementioned four algorithms can be transferred to the multiclass classification task of COVID-19 diagnosis.

On the COVID-19 datasets, several recent publications reported promising results. For example, Cohen et al. [11] proposed a COVID severity score network (CSSNet), which achieved a mean absolute error (MAE) of 1.14 on geographic extent score and an MAE of 0.78 on lung opacity score. Li et al. [12] developed a fully automatic model (COVNet) to detect COVID-19 using chest CT and evaluated its performance. Wang et al. [13] proposed a 3D deep convolutional neural network to detect COVID-19 (DeCovNet). Zhang et al. [14] proposed a seven-layer convolutional neural network for COVID-19 diagnosis (7L-CCD). Their performance achieved an accuracy of  $94.03 \pm 0.80$  for the binary classification task (COVID-19 against healthy subjects). Ko et al. [15] proposed a fast-track COVID-19 classification network (FCONet). For the sake of the page limit, the details of those methods are not described, but we shall compare our method with those state-of-the-art methods in the following sections. Wang et al. [16] presented a CSHNet via transfer learning and discriminant correlation analysis.

Our study’s inspiration is to improve recognition performances of COVID-19 infection in CCT images by developing a novel deep neural network, PSSPNN, short for PatchShuffle stochastic pooling neural network. Our contributions entail the following five angles:

- (i) The “ $n$ -conv stochastic pooling module (NCSPM)” is proposed, which comprises  $n$ -times repetitions of

convolution layers and batch normalization layers, followed by stochastic pooling

- (ii) A novel “stochastic pooling neural network (SPNN)” is proposed, the structure of which is inspired by VGG-16
- (iii) A more advanced neural network, PatchShuffle SPNN (PSSPNN), is proposed where PatchShuffle is introduced as the regularization term in the loss function of SPNN
- (iv) An improved multiple-way data augmentation is utilized to help the network avoid overfitting
- (v) Grad-CAM is used to show the explainable heatmap, which displays association with lung lesions

## 2. Dataset

This retrospective study was exempt by Institutional Review Board of local hospitals. Four types of CCT were used: (i) COVID-19-positive patients, (ii) community-acquired pneumonia (CAP), (iii) second pulmonary tuberculosis (SPT), and (iv) healthy control (HC). Three diseased classes (COVID-19, CAP, and SPT) were chosen since they are all infectious diseases of the chest regions. We intend to include the fifth category (chest tumors) in our future studies.

For each subject,  $n(k)$  slices of CCT were chosen via a slice level selection (SLS) method. For the three diseased groups (COVID-19, CAP, and SPT represented as  $k = \{1, 2, 3\}$ ), the slice displaying the largest number of lesions and size was chosen. For HC subjects ( $k = 4$ ), any slice within the 3D image was randomly chosen. The slice-to-subject ratio  $\bar{n}$  is defined as

$$\bar{n}(k) = \frac{N_S(k)}{N_P(k)}, k = 1, \dots, 4, \quad (1)$$

where  $N_S$  stands for the number of slices via the SLS method and  $N_P$  is the number of patients.

In all, we enrolled 521 subjects and produced 1164 slice images using the SLS method. Table 1 lists the demographics of the four-category subject cohort with the values of triplets  $[\bar{n}, N_P, N_S]$ , where  $\bar{n}$  of the total set equals to 2.23.

Three experienced radiologists (two juniors:  $\mathcal{E}_1$  and  $\mathcal{E}_2$  and one senior:  $\mathcal{E}_3$ ) were convened to curate all the images. Suppose  $x^{\text{CCT}}$  means one CCT scan,  $Y$  stands for the labeling of each individual radiologist. The last labeling  $Y^F$  of the CCT scan  $x^{\text{CCT}}$  is obtained by

$$Y^F[x^{\text{CCT}}] = \begin{cases} Y[\mathcal{E}_1, x^{\text{CCT}}] & Y[\mathcal{E}_1, x^{\text{CCT}}] == Y[\mathcal{E}_2, x^{\text{CCT}}] \\ \text{MAV}\{Y^{\text{all}}[x^{\text{CCT}}]\} & \text{otherwise} \end{cases}, \quad (2)$$

TABLE 1: Subjects and images of four categories.

Class index	Class name	$\bar{n}$	$N_p$	$N_S$
1	COVID-19	2.27	125	284
2	CAP	2.28	123	281
3	SPT	2.18	134	293
4	HC	2.20	139	306
Total		2.23	521	1164

where MAV denotes majority voting and  $Y^{\text{all}}$  the labeling of all radiologists, *viz.*,

$$Y^{\text{all}}[x^{\text{CCT}}] = [Y(\mathcal{E}_1, x^{\text{CCT}}), Y(\mathcal{E}_2, x^{\text{CCT}}), Y(\mathcal{E}_3, x^{\text{CCT}})]. \quad (3)$$

The above two equations mean that in situations of disagreement between the analyses of two junior radiologists ( $\mathcal{E}_1, \mathcal{E}_2$ ), we consult a senior radiologist ( $\mathcal{E}_3$ ) to reach a MAV-type consensus. Data is available on request due to privacy/ethical restrictions.

### 3. Methodology

Table 2 gives the abbreviation and full meanings in this study. Section 3.1 shows the preprocessing procedure. Sections 3.2–3.5 offer four improvements. Finally, Section 3.6 gives the implementation, measure indicators, and explainable technology used in our method.

**3.1. Preprocessing.** The original raw dataset contained  $|V|$  slice images  $\{v_a(i), i = 1, 2, \dots, |V|\}$ . The size of each image was  $\text{size}[v_a(i)] = 1024 \times 1024 \times 3$ . Figure 1 presents the pipeline for preprocessing of this dataset.

First, the color CCT images of four classes were converted into grayscale by retaining the luminance channel and obtaining the grayscale data set  $V_B$ :

$$\begin{aligned} V_B &= O_{\text{gray}}(V_A) \\ &= \{v_b(1), v_b(2), \dots, v_b(i), \dots, v_b(|V|)\}, \end{aligned} \quad (4)$$

where  $O_{\text{gray}}$  means the grayscale operation.

In the second step, the histogram stretching (HS) was utilized to increase the contrast of all images. Take the  $i$ -th image  $v_b(i), i = 1, 2, \dots, |V|$  as an example; its minimum and maximum grayscale values  $v_b^l(i)$  and  $v_b^h(i)$  were calculated as follows:

$$\begin{cases} v_b^l(i) = \min_{w=1}^{W_B} \min_{h=1}^{H_B} \min_{c=1}^{C_B} v_b(i | w, h, c), \\ v_b^h(i) = \max_{w=1}^{W_B} \max_{h=1}^{H_B} \max_{c=1}^{C_B} v_b(i | w, h, c). \end{cases} \quad (5)$$

Here,  $(w, h, c)$  means the index of width, height, and channel directions along image  $v_b(i)$ , respectively.  $(W_B, H_B, C_B)$  means the maximum values of width, height, and channel

TABLE 2: Abbreviation and full name.

Abbreviation	Full name
AP	Average pooling
CAP	Community-acquired pneumonia
CCT	Chest computed tomography
CI	Chest imaging
CXR	Chest X-ray
CUS	Chest ultrasound
DPD	Discrete probability distribution
FCL	Fully connected layer
FM	Feature map
FMS	Feature map size
HS	Hyperparameter size
L2P	l2-norm pooling
MA	Microaveraged
MAV	Majority voting
MDA	Multiple-way data augmentation
MP	Max pooling
NCSPM	n-conv stochastic pooling module
NWL	Number of weighted layers
SAPN	Salt-and-pepper noise
SC	Strided convolution
SLS	Slice level selection
SP	Stochastic pooling
SPNN	Stochastic pooling neural network
SPT	Second pulmonary tuberculosis
TCM	Test confusion matrix

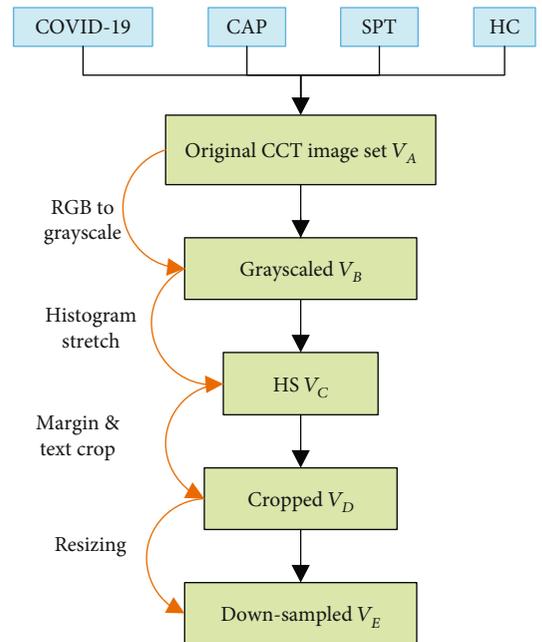


FIGURE 1: Illustration of preprocessing.

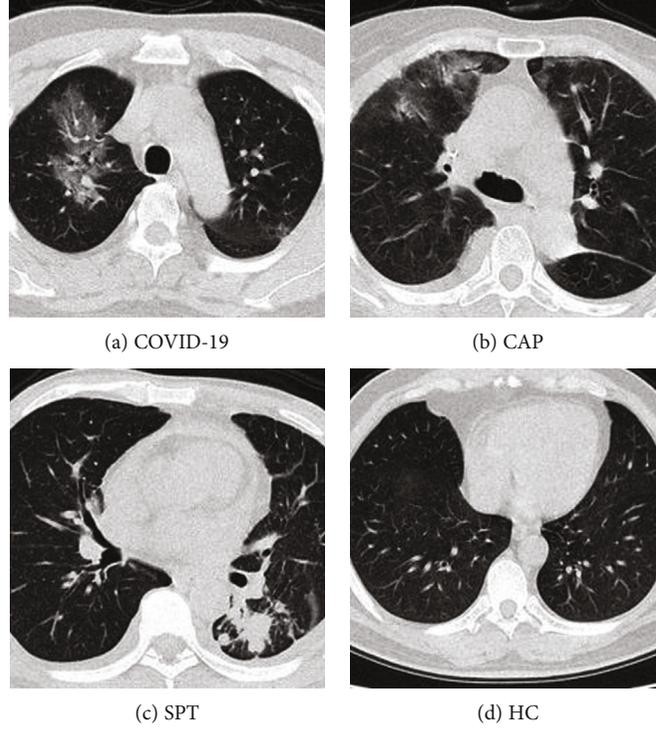


FIGURE 2: Samples of four categories (three diseased and one healthy).

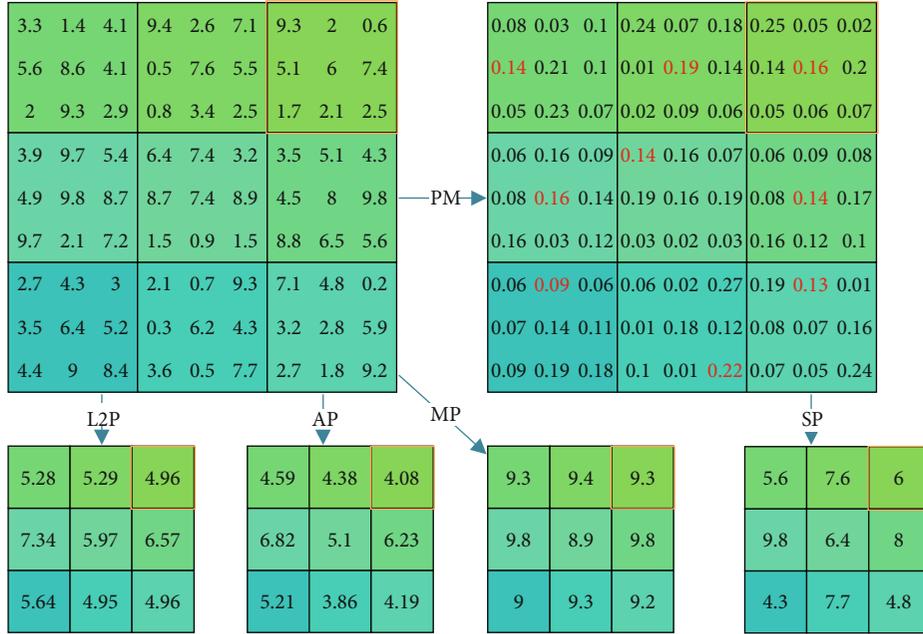


FIGURE 3: Comparison of four different pooling techniques (red rectangle indicates the example discussed in Section 3.2).

to the image set  $V_B$ . The new histogram stretched image set  $V_C$  was calculated as follows:

$$V_C = O_{HS}(V_B) = \left\{ v_c(i) \stackrel{\text{def}}{=} \frac{v_b(i) - v_b^l(i)}{v_b^h(i) - v_b^l(i)} \right\}, \quad (6)$$

where  $O_{HS}$  stands for the HS operation.

In the third step, cropping was performed to remove the checkup bed at the bottom area and eliminate the texts at the margin regions. Cropped dataset  $V_D$  is yielded as

$$V_D = O_{\text{crop}}(V_C, [a_1, a_2, a_3, a_4]) = \{v_d(1), v_d(2), \dots, v_d(i), \dots, v_d(|V|)\}, \quad (7)$$

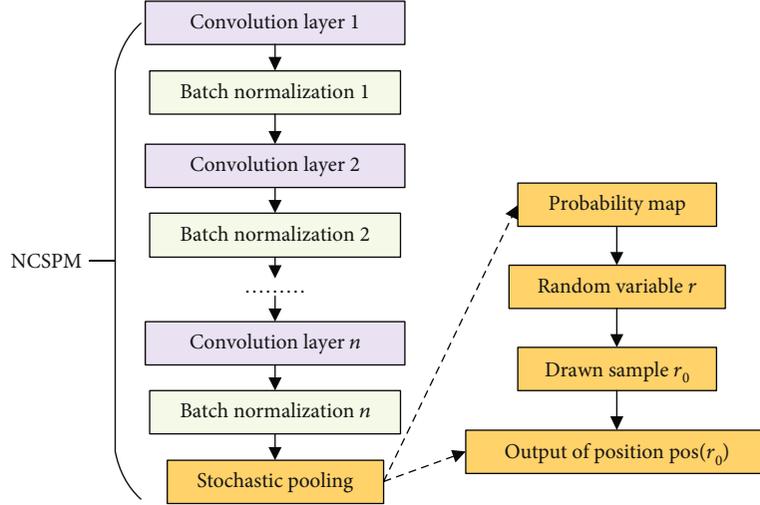


FIGURE 4: Schematic of proposed NCSPM.

where  $O_{\text{crop}}$  represents crop operation. Parameters  $(a_1, a_2, a_3, a_4)$  mean pixels to be cropped from four directions of the top, bottom, left, and right, respectively (unit: pixel).

In the fourth step, each image was downsampled to a size of  $[W_E, H_E]$ , obtaining the resized image set  $V_E$  as

$$V_E = O_{\text{DS}}(V_D, [W_E, H_E]) = \{v_e(1), v_e(2), \dots, v_e(i), \dots, v_e(|V|)\} \quad (8)$$

where  $O_{\text{DS}} : a \mapsto b$  represents the downsampling (DS) procedure, in which  $b$  stands for the downsampled image of the raw image  $a$ .

Figure 2 displays example images of the four categories, in which three are diseased, and one is healthy. The original size of each image in  $V_A$  is  $1024 \times 1024 \times 3$ , and the final pre-processed image is  $256 \times 256 \times 1$ .

**3.2. Improvement I:  $n$ -conv Stochastic Pooling Module.** First, stochastic pooling (SP) [17] was introduced. In the standard convolutional neural networks, pooling is an essential component after each convolution layer, which was applied to reduce the size of feature maps (FMs). SP was shown to give better performance than average pooling and max pooling in recent publications [18–21]. Recently, strided convolution (SC) is commonly used, which also can shrink the FMs [22, 23]. Nevertheless, SC could be considered a simple pooling method, which always outputs the region’s fixed-position value [24].

Suppose we have a postconvolution FM  $\mathbb{F} = f_{ij}(i = 1, \dots, M \times P, j = 1, \dots, N \times Q)$ . The FM can be separated into  $M \times N$  blocks, in which every block has the size of  $P \times Q$ . Now we focus on the block  $B_{mn} = \{b(x, y), x = 1, \dots, P, y = 1, \dots, Q\}$  which stands for the  $m$ -th row and  $n$ -th column blocks.

TABLE 3: Structure of proposed 11-layer SPNN.

Index	Name	NWL	HS	FMS
1	Input			$256 \times 256 \times 1$
2	NCSPM-1	2	$[3 \times 3, 32] \times 2$	$128 \times 128 \times 32$
3	NCSPM-2	2	$[3 \times 3, 32] \times 2$	$64 \times 64 \times 32$
4	NCSPM-3	2	$[3 \times 3, 64] \times 2$	$32 \times 32 \times 64$
5	NCSPM-4	2	$[3 \times 3, 64] \times 2$	$16 \times 16 \times 64$
6	NCSPM-5	1	$[3 \times 3, 128] \times 1$	$8 \times 8 \times 128$
7	Flatten			$1 \times 1 \times 8192$
8	FCL-1	1	$140 \times 8192, 140 \times 1$	$1 \times 1 \times 140$
9	FCL-2	1	$4 \times 140, 4 \times 1$	$1 \times 1 \times 4$
10	Softmax			$1 \times 1 \times 4$

The strided convolution (SC) traverses the input activation map with the strides, which equals the size of the block  $(P, Q)$ , so here its output is set to

$$B_{mn}^{\text{SC}} = b(1, 1). \quad (9)$$

The l2-norm pooling (L2P), average pooling (AP), and max pooling (MP) generate the l2-norm value, average value, and maximum value within the block  $B_{mn}$ , respectively.

$$\begin{cases} B_{mn}^{\text{L2P}} = \sqrt{\frac{\sum_{x=1}^P \sum_{y=1}^Q b^2(x, y)}{P \times Q}}, \\ B_{mn}^{\text{AP}} = \frac{1}{P \times Q} \sum_{x=1}^P \sum_{y=1}^Q b(x, y), \\ B_{mn}^{\text{MP}} = \max_{x=1}^P \max_{y=1}^Q b(x, y). \end{cases} \quad (10)$$

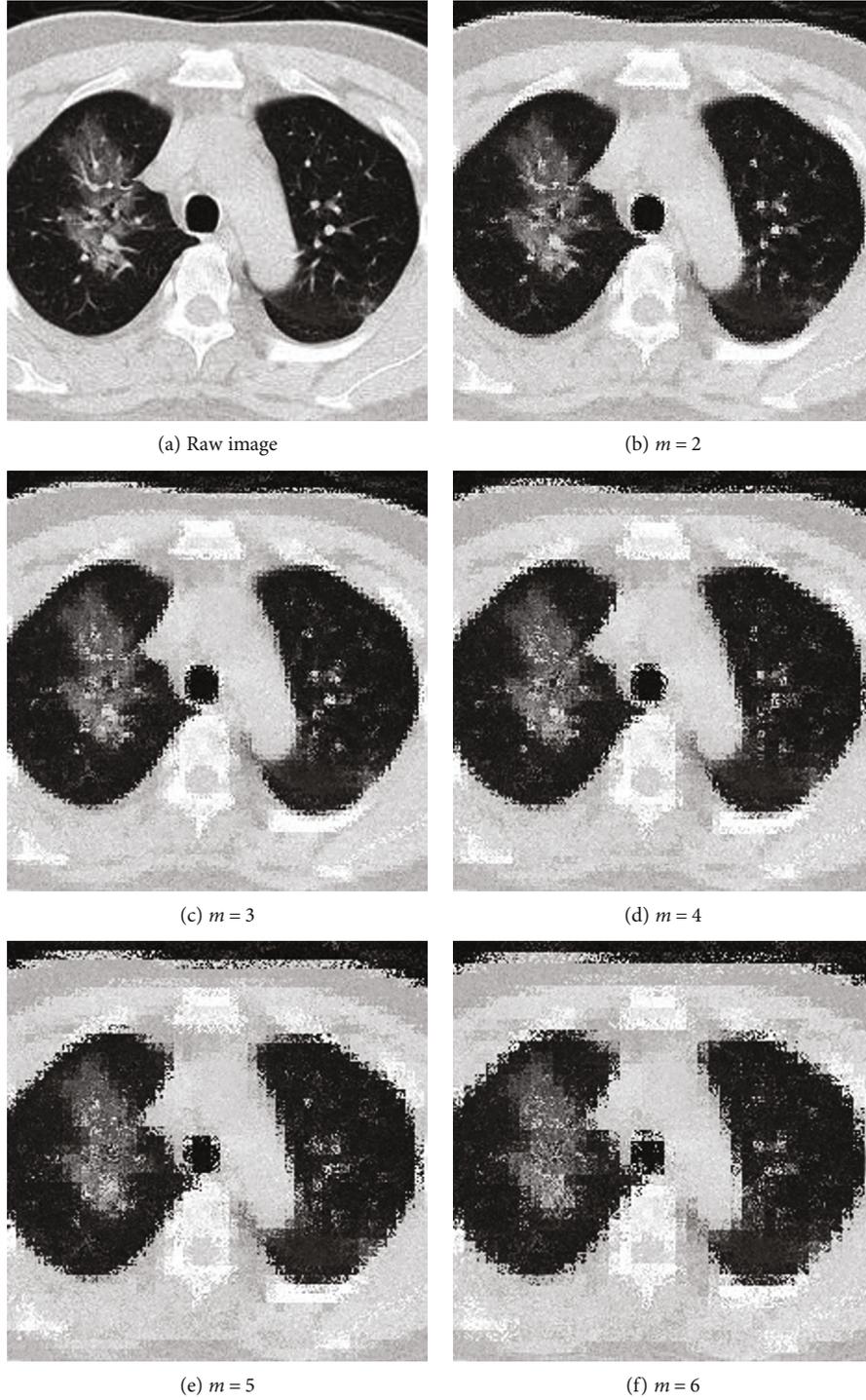


FIGURE 5: Illustration of PatchShuffle on a grayscale COVID-19 image.

The SP provides a solution to the shortcomings of AP and MP. The AP outputs the average, so it will downscale the largest value, where the important features may sit on. On the other hand, MP reserves the maximum value but worsens the overfitting problem. SP is a three-step procedure. First, it generates the probability map (PM) for each entry in the block  $B_{mn}$ .

$$\begin{cases} p_m(x, y) = \frac{b(x, y)}{\sum_{x=1}^P \sum_{y=1}^Q b(x, y)}, \\ s.t. \sum_{x=1}^P \sum_{y=1}^Q p_m(x, y) = 1, \end{cases} \quad (11)$$

where  $p_m(x, y)$  stands for the PM value at pixel  $(x, y)$ .



FIGURE 6: Illustration of PatchShuffle on a colorful house image.

In the second step, create a random variable  $r$  that takes the discrete probability distribution (DPD) as

$$\left\{ \begin{array}{llll} \Pr [r = (1, 1)] = p_m(1, 1) & \Pr [r = (1, 2)] = p_m(1, 2) & \cdots & \Pr [r = (1, Q)] = p_m(1, Q), \\ \Pr [r = (2, 1)] = p_m(2, 1) & \Pr [r = (2, 2)] = p_m(2, 2) & \cdots & \Pr [r = (2, Q)] = p_m(2, Q), \\ & \cdots & \cdots & \cdots \\ \Pr [r = (P, 1)] = p_m(P, 1) & \Pr [r = (P, 2)] = p_m(P, 2) & \cdots & \Pr [r = (P, Q)] = p_m(P, Q), \end{array} \right. \quad (12)$$

where  $\text{Pr}$  represents the probability.

In the third step, a sample  $r_0$  is drawn from the random variable  $r$ , and the corresponding position  $\text{pos}(r_0) = (x_{r_0}, y_{r_0})$ . Then, the output of SP is at location  $\text{pos}(r_0)$ , namely,

$$B_{mn}^{\text{SP}} = b(x_{r_0}, y_{r_0}). \quad (13)$$

Figure 3 presents the comparison of four different pooling techniques. The top left shows the raw FM in which the pooling will take place at a  $3 \times 3$  kernel. If we take the top-right block (in an orange rectangle) as an example, the L2P outputs 4.96, while AP and MP output 4.08 and 9.3, respectively. For the SP method, it will first generate the PM and then sample a position based on the PM (see the red fonts), and thus, SP outputs the value of 6.

A new “n-conv stochastic pooling module” (NCSPM) is proposed in this study based on the SP layer discussed in previous paragraphs. The NCSPM entails  $n$ -repetitions of a conv layer and a batch normalization layer, followed by an SP layer. Figure 4 shows the schematic of the proposed NCSPM module. In this study,  $n = 1\sqrt{2}$ , since we experimented using  $n = 3$ , but the performance using  $n = 3$  did not improve.

**3.3. Improvement II: Stochastic Pooling Neural Network.** The second improvement of this study is to propose a stochastic pooling neural network (SPNN), whose structure was inspired by VGG-16 [25]. In VGG-16, the network used small kernels instead of large kernels and always used  $2 \times 2$  filters with a stride of 2 for pooling. In the end, VGG-16 has two fully connected layers (FCLs).

This proposed SPNN will follow the same structure design of VGG-16 but using the NCSPM module to replace the convolution block in VGG-16. The details of SPNN are shown in Table 3, where NWL means the number of weighted layers, HS is the hyperparameter size, and FMS is the feature map size.

Compared to ordinary CNN, the advantages of SPNN are two folds: (i) SPNN helps prevent overfitting; (ii) SPNN is parameter-free. (iii) SPNN can be easily combined with other advanced techniques, such as batch normalization and dropout. In total, we create this 11-layer SPNN. We have attempted to insert more NCSPMs or more FCLs, which does not show performance improvement but more computation burden. The structure of the proposed model is summarized in Table 3. The  $[c_1 \times c_1, c_2] \times c_3$  related to NCSPM stands for  $c_3$  repetitions of  $c_2$  filters with size of  $c_1 \times c_1$ . For the FCL, the  $c_1 \times c_2, c_3 \times c_4$  stands for a weight matrix is with size of  $c_1 \times c_2$  and a bias matrix is with size of  $c_3 \times c_4$ . In the last column of Table 3, the format of  $c_1 \times c_2 \times c_3$  represents the feature map’s size in three dimensions: height, width, and channel. Directly using transfer learning is another alternative.

In this study, we chose to create a custom neural network by designing its structure and training the whole network using our own data. The reason is some reports have shown this “built your own network from scratch” can achieve better performance than transfer learning [26, 27].

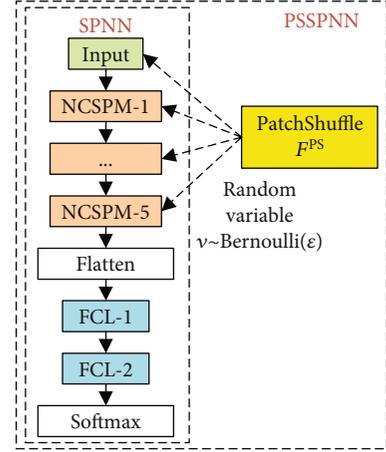


FIGURE 7: Build PSSPNN from SPNN.

**3.4. Improvement III: PatchShuffle SPNN.** Kang et al. [28] presented a new PatchShuffle method. In each minibatch, images and feature maps undergo a transformation such that pixels with that patch are shuffled. By generating fake images/feature maps with interior order-less patches, PatchShuffle creates local variations and reduces the possibility of the AI model overfitting. Therefore, PatchShuffle is a beneficial supplement to various existing training regularization methods [28].

Assume there is a matrix  $X$  of  $M \times M$  entries. A random variable  $v$  controls whether the matrix  $X$  to be PatchShuffled or not. The random variable  $v$  obeys the Bernoulli distribution

$$v \sim \text{Bernoulli}(\epsilon). \quad (14)$$

Namely,  $v = 1$  with probability  $\epsilon$ , and  $v = 0$  with probability  $1 - \epsilon$ . The resulted matrix  $\hat{X}$  is written as

$$\hat{X} = (1 - v)X + vF^{\text{PS}}(X), \quad (15)$$

where  $F^{\text{PS}}$  is the PatchShuffle operation. Suppose the size of each patch is  $m \times m$ , we can express the matrix  $X$  as

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1,M/m} \\ x_{21} & x_{22} & \cdots & x_{2,M/m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M/m,1} & x_{M/m,2} & \cdots & x_{M/m,M/m} \end{bmatrix}, \quad (16)$$

where  $x_{ij}$  stands for a nonoverlapping patch at  $i$ -th row and  $j$ -th column. The PatchShuffle transformation works on every patch.

$$F^{\text{PS}}(X) = \left\{ F^{\text{PS}}(x_{ij}), i = 1, \dots, \frac{M}{m}, j = 1, \dots, \frac{M}{m} \right\}. \quad (17)$$

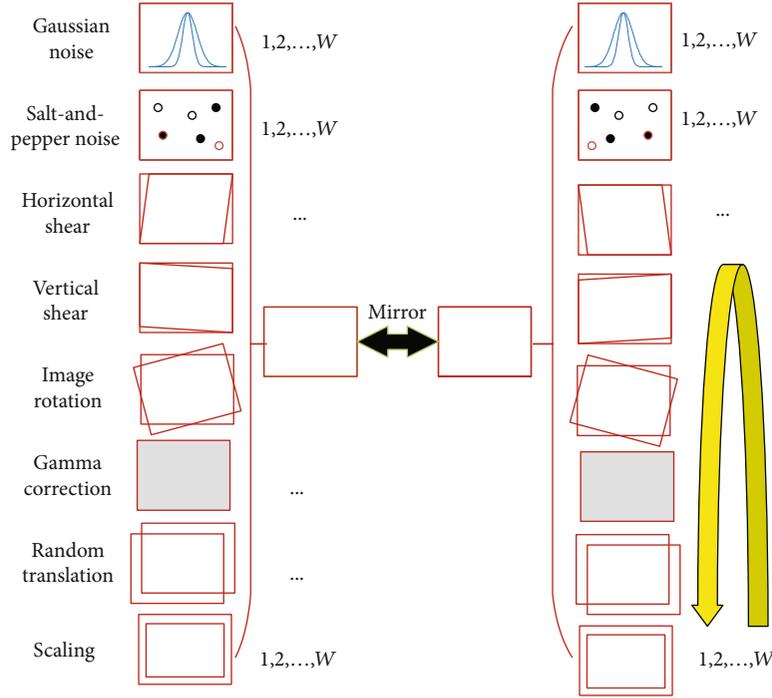


FIGURE 8: The proposed 16-way improved data augmentation method.

TABLE 4: Pseudocode of proposed 16-way improved data augmentation.

Input	Raw image $v(i)$
Step 1	Eight geometric or photometric DA transforms were utilized on raw image $v(i)$ .
Step 2	A horizontal mirror image is generated.
Step 3	The raw image $v(i)$ , the mirrored image, the above 8-way DA results of the raw image, and the 8-way DA results of the horizontal mirrored image are combined to form a dataset $\mathbb{D}(i)$ .
Output	Enhanced dataset of raw image $\mathbb{D}(i)$

The shuffled patch is formulated as

$$F^{\text{PS}}(x_{ij}) = e_{ij} \times x_{ij} \times e'_{ij}, \quad (18)$$

where  $e_{ij}$  is the row permutation matrix and  $e'_{ij}$  is the column permutation matrix [29]. In practice, a randomly shuffle operation is used to replace the row and column permutation operation. Each patch will undergo one of the  $m^2!$  possible permutations.

We proposed integrating PatchShuffle into our SPNN, and this new network model is named as PatchShuffle stochastic pooling neural network (PSSPNN). The PatchShuffle operation acts on both the input image layer (see grayscale image Figure 5 and colorful image Figure 6 with different values of  $m$ ) and the feature maps of all the convolutional layers (9 conv layers from NCSPM-1 to NCSPM-5).

The schematics of building PSSPNN from the SPNN are drawn in Figure 7, where either inputs or feature maps are randomly selected to undergo the PatchShuffle operation. To reach the best bias-variance trade-off, only a small per-

centage ( $\epsilon$ ) of the images/feature maps will undergo  $F^{\text{PS}}$  operation.

For simplicity, we consider the PatchShuffle images as an example, and the training loss function  $\ell$  of the proposed PSSPNN is written as

$$\ell^{\text{PSSPNN}}(\mathbf{X}, y, \mathcal{W}) = (1 - \nu)\ell(\mathbf{X}, y, \mathcal{W}) + \nu\ell[F^{\text{PS}}(\mathbf{X}), y, \mathcal{W}], \quad (19)$$

where  $\ell$  stands for the ordinary loss function and  $\ell^{\text{PSSPNN}}$  the loss function of PSSPNN.  $\mathbf{X}$  represents the original images and  $F^{\text{PS}}(\mathbf{X})$  the PatchShuffled images. The label is symbolized as  $y$ , and the weights are symbolized as  $\mathcal{W}$ .

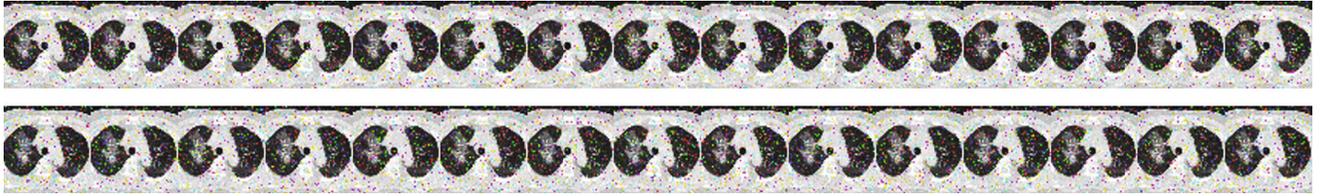
Considering the extreme situations when  $\nu = 0 \vee 1$ , we have

$$\ell^{\text{PSSPNN}}(\mathbf{X}, y, \mathcal{W}) = \begin{cases} \ell(\mathbf{X}, y, \mathcal{W}) & \nu = 0, \\ \ell[F^{\text{PS}}(\mathbf{X}), y, \mathcal{W}] & \nu = 1, \end{cases} \quad (20)$$

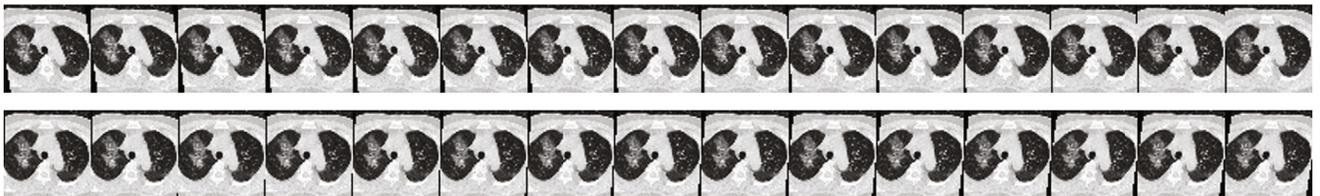
which means the loss function degrades to ordinary loss function when  $\nu = 0$ , and meanwhile, the loss function equals



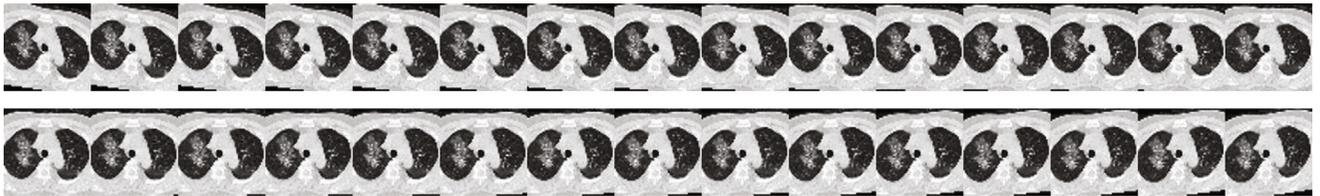
(a) Gaussian noise



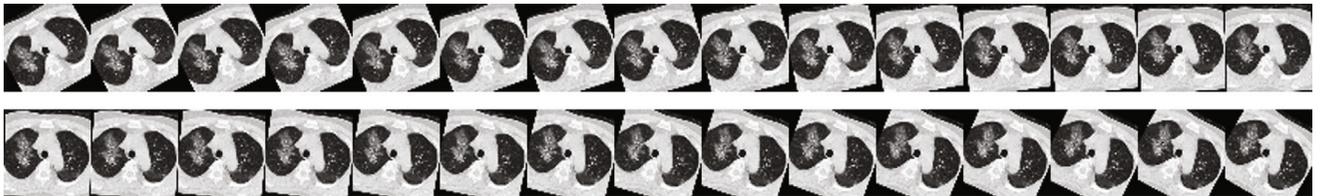
(b) Salt-and-pepper noise



(c) Horizontal shear



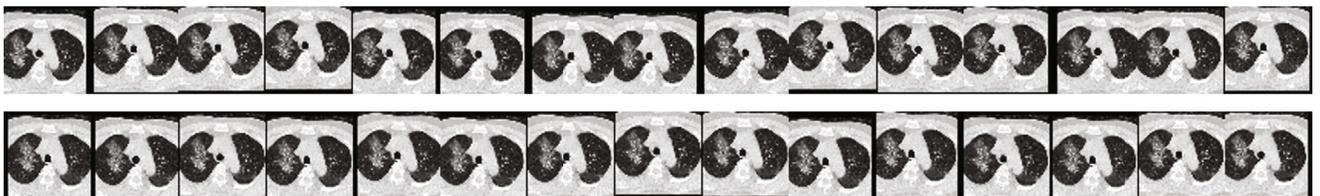
(d) Vertical shear



(e) Rotation

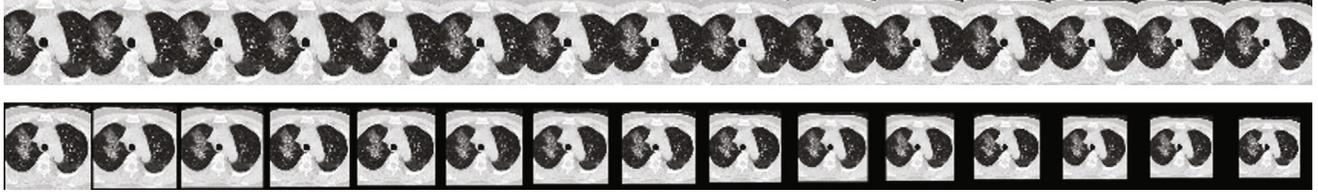


(f) Gamma correction



(g) Random translation

FIGURE 9: Continued.



(h) Scaling

FIGURE 9: Proposed 16-way data augmentation results.

TABLE 5: Dataset splitting.

	Nontest (10-fold cross-validation)	Test (10 runs)	Total
COVID-19	$ V_1^{\text{nntest}}  = 227$	$ V_1^{\text{test}}  = 57$	$ V_1  = 284$
CAP	$ V_2^{\text{nntest}}  = 225$	$ V_2^{\text{test}}  = 56$	$ V_2  = 281$
SPT	$ V_3^{\text{nntest}}  = 234$	$ V_3^{\text{test}}  = 59$	$ V_3  = 293$
HC	$ V_4^{\text{nntest}}  = 245$	$ V_4^{\text{test}}  = 61$	$ V_4  = 306$

to training all images PatchShuffled when  $v = 1$ . Taking mathematical expectation of  $v$ , equation (19) turns to

$$\frac{1}{1-\varepsilon} \mathbb{E}_v \ell^{\text{PSSPNN}}(\mathbf{X}, y, \mathcal{W}) = \ell(\mathbf{X}, y, \mathcal{W}) + \frac{\varepsilon}{1-\varepsilon} \ell[F^{\text{PS}}(\mathbf{X}), y, \mathcal{W}], \quad (21)$$

where  $\varepsilon/1 - \varepsilon \ell[F^{\text{PS}}(\mathbf{X}), y, \mathcal{W}]$  acts as a regularization term.

**3.5. Improvement IV: Improved Multiple-Way Data Augmentation.** This small four-category dataset makes our AI model prone to overfitting. In order to alleviate the overfitting and handle the low sample-size problem, the multiple-way data augmentation (MDA) [30] method was chosen and further improved. In the original 14-way MDA [30], the authors used seven different data augmentation (DA) techniques to the raw image and its horizontal image. Their seven DA techniques are as follows: noise injection, horizontal shear, vertical shear, rotation, Gamma correction, scaling, and translation.

Figure 8 shows a 16-way data augmentation method. The difference between the proposed 16-way DA with the traditional 14-way DA is that we add the salt-and-pepper noise (SAPN). Although the SAPN defies intuition as it never takes place in realistic CCT images, we found that it can increase performance. The same observation was reported by Li et al. [31], where the authors used salt and pepper noise for the identification of early esophageal cancer. Table 4 shows the pseudocode of this proposed 16-way improved data augmentation.

$$v(i) \mapsto \mathbb{D}(i) = \mathbb{S} \left\{ \underbrace{\mathbb{S}_{k=1, \dots, 8} f_k^{\text{DA}}[v(i)]}_W, \underbrace{\mathbb{S}_{k=1, \dots, 8} f_k^{\text{DA}}\{M[v(i)]\}}_W \right\}, \quad (22)$$

TABLE 6: Comparison of SPNN with four standard CNNs.

Model	Class	Sen	Prc	F1
SC-CNN	C1	93.51	93.35	93.43
	C2	86.79	91.87	89.26
	C3	94.41	91.01	92.68
	C4	94.59	93.37	93.97
	MA			92.40
AP-CNN	C1	92.86	93.02	92.94
	C2	95.76	92.93	94.32
	C3	94.26	93.95	94.11
	C4	93.22	93.22	93.22
	MA			93.18
SPNN (ours)	C1	98.07	94.59	96.30
	C2	91.79	95.19	93.45
	C3	94.92	94.92	94.92
	C4	95.25	95.40	95.32
	MA			95.02
L2P-CNN	C1	88.77	93.36	91.01
	C2	93.21	93.72	93.46
	C3	94.41	92.68	93.53
	C4	93.61	90.63	92.10
	MA			92.53
MP-CNN	C1	95.44	94.77	95.10
	C2	94.64	91.70	93.15
	C3	94.07	93.43	93.75
	C4	91.64	95.72	93.63
	MA			93.91

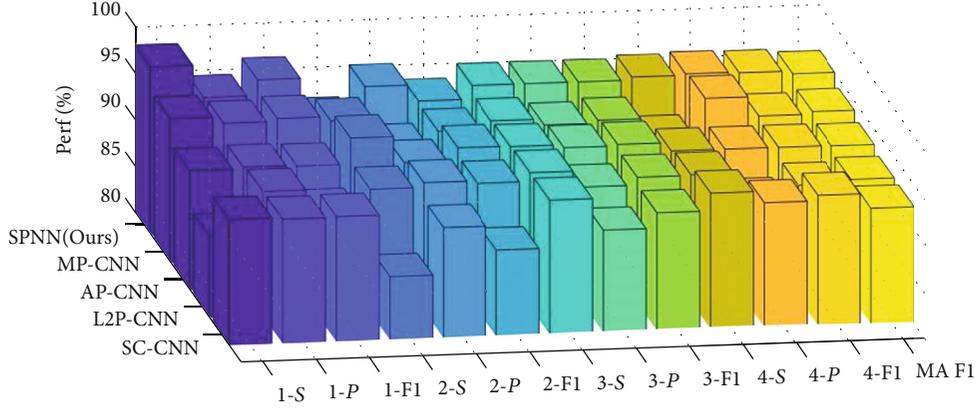


FIGURE 10: 3D bar plot of SPNN against other network models.

where  $W = 30$  in this study. We tested a greater value of  $W$ , but it does not bring about significant improvement. Hence, one image  $v(i)$  will generate  $|\mathbb{D}(i)| = 16W + 2 = 482$  images (including original image), as shown in Figure 8.

*Step 1.* Eight geometric or photometric DA transforms were utilized on raw image  $v(i)$ , as shown in Figure 8. We use  $f_k^{\text{DA}}, k = 1, \dots, 8$  to denote each DA operation. Note each DA operation  $f_k^{\text{DA}}$  will yield  $W$  new images. So, for a given image  $v(i)$ , we will produce an enhanced dataset  $\mathbb{S}_{k=1, \dots, 8} f_k^{\text{DA}}[v(i)]$ , where  $\mathbb{S}$  stands for concatenation function.

*Step 2.* Horizontal mirror image is generated as  $M[v(i)]$ , where  $M$  means horizontal mirror function.

*Step 3.* The raw image  $v(i)$ , the mirrored image  $M[v(i)]$ , all the above 8-way DA results of raw image  $\mathbb{S}_{k=1, \dots, 8} f_k^{\text{DA}}[v(i)]$ , and 8-way DA results of horizontal mirrored image  $\mathbb{S}_{k=1, \dots, 8} f_k^{\text{DA}}\{M[v(i)]\}$  are combined. Mathematically, one training image  $v(i)$  will generate a dataset  $\mathbb{D}(i)$ , which contains  $16W + 1$  new images.

Taking Figure 2(a) as an example raw image, Figure 9 shows the 8-way DA results, i.e.,  $f_k^{\text{DA}}[v(i)], k = 1, \dots, 8$ . Due to the page limit, the mirror image and its corresponding 8-way DA results are not shown here.

**3.6. Implementation, Measure, and Explainability.** Table 5 itemizes the nontest and test sets for each category. The whole dataset  $V$  contains four nonoverlapping categories  $V = \{V_1, V_2, V_3, V_4\}$ . For each category, the dataset will be split into nontest set and test set  $V_k \rightarrow \{V_k^{\text{nontest}}, V_k^{\text{test}}\}, k = 1, 2, 3, 4$ .

$$V = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = [V^{\text{nontest}} \quad V^{\text{test}}] = \begin{bmatrix} V_1^{\text{nontest}} & V_1^{\text{test}} \\ V_2^{\text{nontest}} & V_2^{\text{test}} \\ V_3^{\text{nontest}} & V_3^{\text{test}} \\ V_4^{\text{nontest}} & V_4^{\text{test}} \end{bmatrix}. \quad (23)$$

TABLE 7: Hyperparameter optimization of PSSPNN in terms of microaveraged F1 score.

$\epsilon$	Patch size			
	$1 \times 2$	$2 \times 2$	$2 \times 4$	$3 \times 3$
0.01	95.19	95.54	95.15	94.98
0.05	95.49	95.79	95.45	95.11
0.10	95.24	95.28	95.54	95.41
0.15	95.24	95.11	95.28	95.06
0.20	95.06	95.15	94.85	94.72

Our experiment entails two phases. At phase I, 10-fold cross-validation was used for validation on the nontest set to select the best hyperparameters and best network structure. The 16-way DA was used on the training set of 10-fold cross-validation. The hyperparameter of the proposed PSSPNN was determined over the nontest set  $V^{\text{nontest}}$ . Afterward at phase II, we train our model using the nontest set  $V^{\text{nontest}}$  10 times with different initial seeds and attain the test results over the test set  $V^{\text{test}}$ . After combining the  $R^{\text{test}}$  runs, we attain a summation of the test confusion matrix (TCM)  $D^{\text{test}}$ . Table 5 shows the dataset splitting, where  $|x|$  stands for the number of elements in the dataset  $x$ .

The ideal TCM is a diagonal matrix with the form of

$$D_{\text{ideal}}^{\text{test}} = R^{\text{test}} \times \begin{bmatrix} |V_1^{\text{test}}| & 0 & 0 & 0 \\ 0 & |V_2^{\text{test}}| & 0 & 0 \\ 0 & 0 & |V_3^{\text{test}}| & 0 \\ 0 & 0 & 0 & |V_4^{\text{test}}| \end{bmatrix}, \quad (24)$$

in which all the off-diagonal elements are zero, meaning no prediction errors. In realistic scenarios, the AI model will make errors, and the performance is calculated per category. For each class  $z = 1, \dots, 4$ , we set the label of that class as *positive*, and the labels of all the rest classes as *negative*. Three

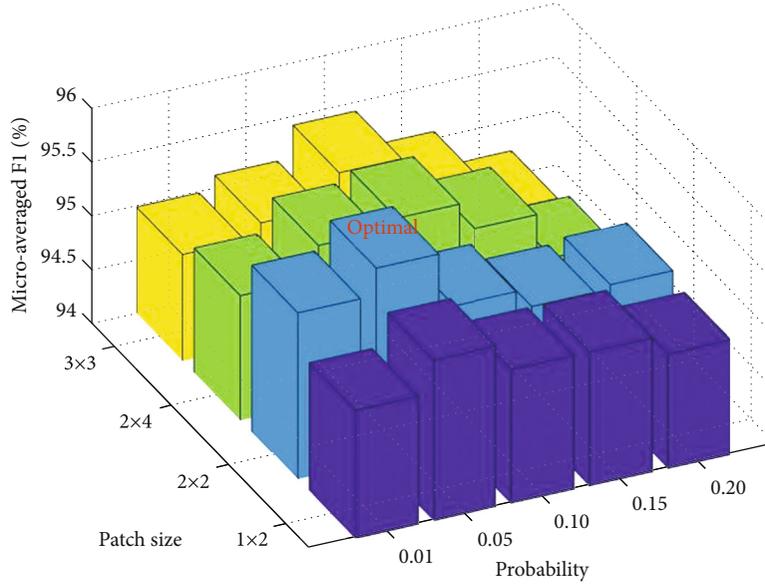


FIGURE 11: 3D bar chart of microaveraged F1 against patch size and probability.

performance metrics (sensitivity, precision, and F1 score) per category are defined below:

$$\begin{cases} \text{Sen}(z) = \frac{\text{TP}(z)}{\text{TP}(z) + \text{FN}(z)}, \\ \text{Prc}(z) = \frac{\text{TP}(z)}{\text{TP}(z) + \text{FP}(z)}, \\ \text{F1}(z) = \frac{2 * \text{Prc}(z) * \text{Sen}(z)}{\text{Prc}(z) + \text{Sen}(z)}. \end{cases} \quad (25)$$

The performance can be measured over all four categories. The microaveraged (MA) F1 (symbolized as  $F1_\mu$ ) is used since our dataset is slightly unbalanced:

$$F1_\mu = \frac{2 * \text{Pr } c_\mu * \text{Sen}_\mu}{\text{Pr } c_\mu + \text{Sen}_\mu}, \quad (26)$$

where

$$\begin{cases} \text{Sen}_\mu = \frac{\sum_{z=1}^4 \text{TP}(z)}{\sum_{z=1}^4 \text{TP}(z) + \text{FN}(z)}, \\ \text{Pr } c_\mu = \frac{\sum_{z=1}^4 \text{TP}(z)}{\sum_{z=1}^4 \text{TP}(z) + \text{FP}(z)}. \end{cases} \quad (27)$$

Finally, gradient-weighted class activation mapping (Grad-CAM) [32] was employed to provide explanations on how our model makes the decision. It exploits the gradient of the categorization score regarding the convolutional features decided by the deep model to visualize the regions of the image that are the most vital for the image classification task [33]. The output of NCSPM-5 in Table 3 was used for Grad-CAM.

TABLE 8: Comparison of SPNN against PSSPN.

Model	Class	Sen	Prc	F1
SPNN (ours)	C1	98.07	94.59	96.30
	C2	91.79	95.19	93.45
	C3	94.92	94.92	94.92
	C4	95.25	95.40	95.32
	MA			95.02
PSSPN (ours)	C1	97.89	95.06	96.46
	C2	92.86	96.30	94.55
	C3	95.76	95.44	95.60
	C4	96.56	96.40	96.48
	MA			95.79

## 4. Experiments, Results, and Discussions

The experiment was carried out on the programming platform of Matlab 2020b. The programs ran on Windows 10 with 16GB RAM and 2.20GHz Intel Core i7-8750H CPU. The performances are reported over the test set with 10 runs.

**4.1. Comparison of SPNN and Other Pooling Methods.** In the first experiment, we compared the proposed SPNN with four standard CNNs with different pooling methods. The first CNN uses strided convolution in five modules to replace the stochastic pooling. The second to fourth comparison CNN models use L2P, AP, and MP, respectively. Those four baseline methods are called SC-CNN, L2P-CNN, AP-CNN, and MP-CNN, respectively. The results of 10 runs over the test set are shown in Table 6. The bar plot is displayed in Figure 10, where “ $k$ -S,” “ $k$ -P,” and “ $k$ -F1” stand for the sensitivity, precision, and F1 score for category  $k \in \{1, 2, 3, 4\}$ .

The results in Table 6 and Figure 10 are coherent with our expectation that SPNN obtained the best results among all FM reduction approaches. The SPNN arrives at the

TABLE 9: Comparison with state-of-the-art approaches.

Model	Class	Sen	Prc	F1
RCBO [7]	C1	71.93	84.19	77.58
	C2	72.86	72.73	72.79
	C3	73.56	76.41	74.96
	C4	80.66	68.91	74.32
	MA			74.85
RN-50 [9]	C1	87.72	85.03	86.36
	C2	87.68	91.26	89.44
	C3	93.39	89.89	91.60
	C4	84.92	87.65	86.26
	MA			88.41
CSSNet [11]	C1	94.04	92.25	93.14
	C2	93.75	95.11	94.42
	C3	91.36	93.58	92.45
	C4	94.43	92.75	93.58
	MA			93.39
DeCovNet [13]	C1	91.05	90.58	90.81
	C2	93.75	90.99	92.35
	C3	90.51	86.97	88.70
	C4	88.69	95.58	92.01
	MA			90.94
FCONet [15]	C1	92.28	95.64	93.93
	C2	96.79	94.43	95.59
	C3	94.75	95.88	95.31
	C4	94.92	92.94	93.92
	MA			94.68
6L-CNN [8]	C1	72.46	83.94	77.78
	C2	78.93	77.82	78.37
	C3	81.86	75.00	78.28
	C4	89.84	87.54	88.67
	MA			80.94
RN-18 [10]	C1	82.81	82.66	82.73
	C2	81.07	74.43	77.61
	C3	74.24	76.98	75.58
	C4	82.13	86.38	84.20
	MA			80.04
COVNet [12]	C1	89.82	86.63	88.20
	C2	89.82	92.63	91.21
	C3	93.73	90.66	92.17
	C4	87.38	90.96	89.13
	MA			90.17
7L-CCD [14]	C1	89.47	93.58	91.48
	C2	93.93	92.44	93.18
	C3	93.73	95.18	94.45
	C4	95.08	91.34	93.17
	MA			93.09

TABLE 9: Continued.

Model	Class	Sen	Prc	F1
PSSPNN (ours)	C1	97.89	95.06	96.46
	C2	92.86	96.30	94.55
	C3	95.76	95.44	95.60
	C4	96.56	96.40	96.48
	MA			95.79

sensitivities of all four categories are 98.07%, 91.79%, 94.92%, and 95.25%, respectively. The precisions of all four categories are 94.59%, 95.19%, 94.92%, and 95.40%, respectively. The F1-scores of the four categories are 96.30%, 93.45%, 94.92%, and 95.32%, respectively. The overall microaveraged F1 is 95.02%.

In terms of microaveraged F1, the second-best algorithm is MP-CNN, which obtains a microaveraged F1 score of 93.91%. The third best is AP-CNN, with a microaveraged F1 score of 93.18%. The two comparably worst algorithms are SC-CNN and L2P-CNN, with the microaveraged F1 scores of 92.40% and 92.53%, respectively.

SPNN obtains the best results because SP can prevent overfitting [17], which is the main shortcoming of max pooling. On the other hand, AP and L2P will average out the maximum activation values, which will impair the performances of convolutional neural network models. For SC-CNN, it only uses one quarter information of the input FM, and therefore may neglect those greatest values [34]. In all, this proposed SPNN can be regarded as an improved version of vanilla CNN models, where the SP is used to replace traditional MP.

*4.2. PSSPNN versus SPNN.* In this second experiment, we compared our two proposed network models, PSSPNN against SPNN, to validate the effectiveness of PatchShuffle. The results of 10 runs over the test set with different combinations of hyperparameters are shown in Table 7, and the 3D bar chart is shown in Figure 11. The optimal hyperparameter we found from the 10-fold cross-validation of the nontest set is  $\epsilon = 0.05$  and patch size is  $2 \times 2$ , which are coherent with reference [28].

In addition, the PSSPNN with optimal hyperparameter is compared with SPNN. The results are shown in Table 8. From the table, we can observe that PSSPNN provides better F1 values for all four categories and the overall microaverage, which shows the effectiveness of PatchShuffle. The reason is PatchShuffle adds regularization terms [28] in the loss function, and thus can improve the generalization ability of our SPNN model.

*4.3. Comparison to State-of-the-Art Approaches.* We compared our proposed PSSPNN method with 9 state-of-the-art methods: RCBO [7], 6L-CNN [8], RN-50 [9], RN-18 [10], CSSNet [11], COVNet [12], DeCovNet [13], 7L-CNN-CD [14], and FCONet [15]. All the comparison was carried on the same test set of 10 runs. The comparison results are shown in Table 9.

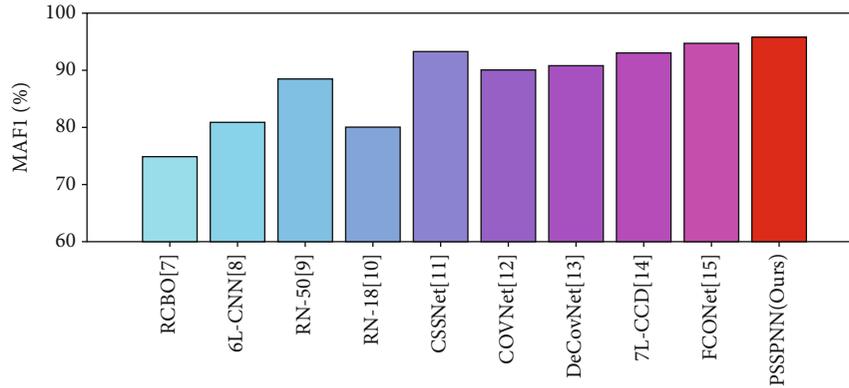


FIGURE 12: Algorithm comparison in terms of MA F1.

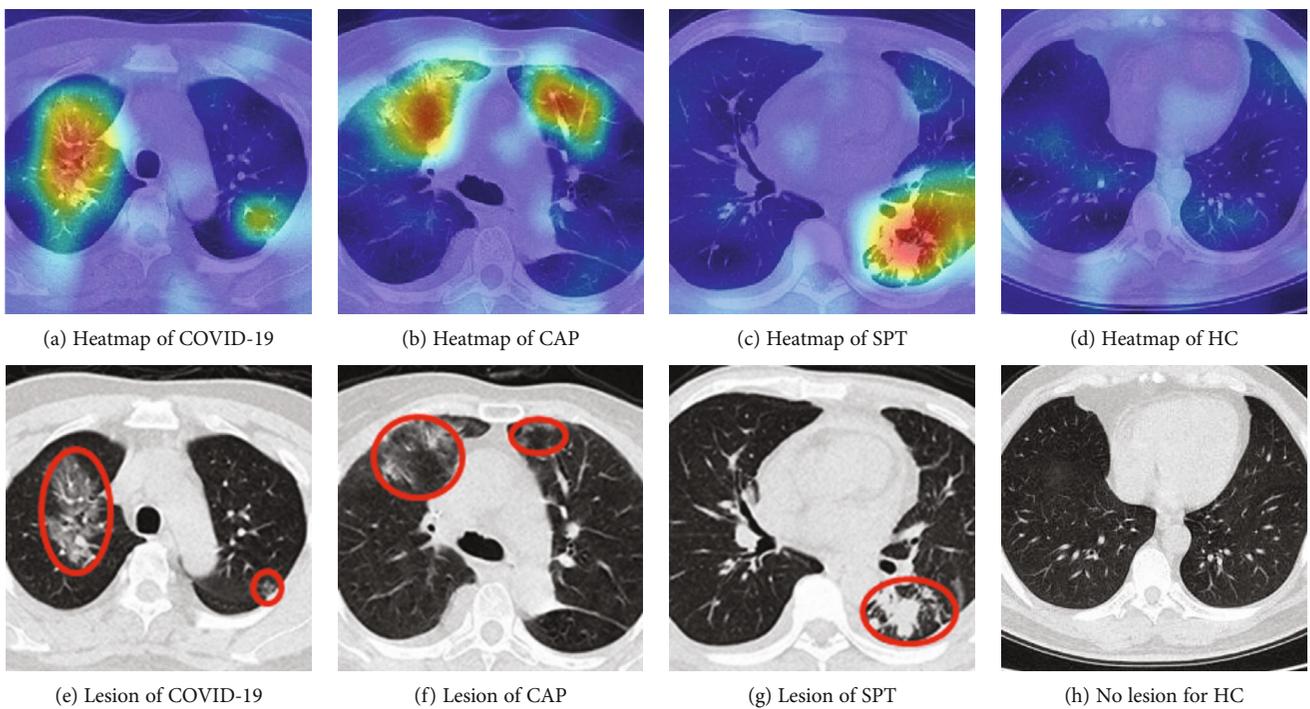


FIGURE 13: Delineation of three diseased samples.

For ease of comparison, Figure 12 only compares the microaveraged F1 (MA F1) score of all algorithms, from which we can observe this proposed PSSPNN achieves the best performance among all the algorithms. This experiment is a simulation-based comparison. In the future, we will apply our algorithm to rigorous clinical testing and verification.

**4.4. Explainability of Proposed PSSPNN.** We take Figure 2 images as examples; the heatmaps of those four images are shown in Figures 13(a)–13(d), and the manual delineation, shown in Figures 13(e)–13(h), delineates the lesions of the three disease samples. Note there are no lesions of healthy control (HC) image. The NCSPM-5 feature map in PSSPNN was used to generate heatmaps with the Grad-CAM approach.

We can observe from Figure 13 that the heatmaps via our PSSPNN model and Grad-CAM can capture the lesions

effectively and meanwhile neglect those nonlesion regions. Traditionally, AI is regarded as a “black box,” which impairs its widespread usage, e.g., the black box properties of traditional AI are problematic for the FDA. Nevertheless, with the help of explainability of modern AI techniques [35], the radiologist and patients will gain confidence in our proposed AI model, as the heatmap provides a clear and understandable interpretation of how AI predicts COVID-19 and other chest infectious disease from healthy subjects, which was also stated in reference [36]. Many new AI-based anatomic pathological systems now pass through FDA approval, such as whole slide images (WSI) [37], since the doctors know the relationships between the diagnosis and the explained answer.

In the future, the explainability of our proposed AI model can be used in patient monitoring [38] and health big data [39]. Some novel network improvement and signal

processing techniques may help our AI model in future researches, such as filters [40, 41], fuzzy [42, 43], edge computing [44], knowledge-aid [45, 46], autofocus [47], graph integration, and cross-domain knowledge exploitation [48–50].

## 5. Conclusion

In this paper, we proposed a PSSPNN, which entails five improvements: (i) proposed NCSPM module, (ii) usage of stochastic pooling, (iii) usage of PatchShuffle, (iv) improved multiple-way data augmentation, and (v) explainability via Grad-CAM. Those five improvements enable our AI model to deliver improved performances compared to 9 state-of-the-art approaches. The 10 runs on the test set showed our algorithm achieved a microaveraged F1 score of 95.79%.

There are three shortcomings of our method, which will be resolved in the future: (i) the dataset currently contains three chest infectious diseases. In the future, we shall try to include more classes of chest diseases, such as thoracic cancer. (ii) Some new network techniques and models are not tested, such as transfer learning, wide network module design, attention mechanism, and graph neural network. Those advanced AI technologies will be studied. (iii) Our model does not go through strict clinical validation, so we will attempt to release our software to hospitals and get feedback from radiologists and consultants.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Authors' Contributions

Shui-Hua Wang and Yin Zhang contributed equally to this paper.

## Acknowledgments

We appreciate Qinghua Zhou for helping us revise our English. This paper is partially supported by the Royal Society International Exchanges Cost Share Award, UK (RP202G0230); Medical Research Council Confidence in Concept Award, UK (MC\_PC\_17171); Hope Foundation for Cancer Research, UK (RM60G0680); British Heart Foundation Accelerator Award, UK; Open fund for Jiangsu Key Laboratory of Advanced Manufacturing Technology (HGAMTL-1703); Guangxi Key Laboratory of Trusted Software (kx201901).

## References

- [1] A. A. Szigiato, M. Anderson, M. Mabon, M. Germain, G. M. Durr, and A. C. Labbe, "Usefulness of prestorage corneal swab culture in the prevention of contaminated corneal tissue in corneal transplantation," *Cornea*, vol. 39, no. 7, pp. 827–833, 2020.
- [2] R. Mögling, A. Meijer, N. Berginc et al., "Delayed laboratory response to covid-19 caused by molecular diagnostic contamination," *Emerging Infectious Diseases*, vol. 26, no. 8, pp. 1944–1946, 2020.
- [3] T. Ai, Z. Yang, H. Hou et al., "Correlation of chest CT and RT-PCR testing for coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases," *Radiology*, vol. 296, no. 2, pp. E32–E40, 2020.
- [4] R. Jain, M. Gupta, S. Taneja, and D. J. Hemanth, "Deep learning based detection and analysis of COVID-19 on chest X-ray images," *Applied Intelligence*, vol. 51, no. 3, pp. 1690–1700, 2021.
- [5] H. Conway, G. Lau, and V. Zochios, "Personalizing invasive mechanical ventilation strategies in coronavirus disease 2019 (COVID-19)-associated lung injury: the utility of lung ultrasound," *Journal of Cardiothoracic and Vascular Anesthesia*, vol. 34, no. 10, pp. 2571–2574, 2020.
- [6] A. K. Goel, D. DiLella, G. Dotsikas, M. Hilts, D. Kwan, and L. Paxton, "Unlocking radiology reporting data: an implementation of synoptic radiology reporting in low-dose CT cancer screening," *Journal of Digital Imaging*, vol. 32, no. 6, pp. 1044–1051, 2019.
- [7] S. Wang, P. Li, P. Chen et al., "Pathological brain detection via wavelet packet Tsallis entropy and real-coded biogeography-based optimization," *Fundamenta Informaticae*, vol. 151, no. 1–4, pp. 275–291, 2017.
- [8] X. Jiang and Y. D. Zhang, "Chinese sign language fingerspelling via six-layer convolutional neural network with leaky rectified linear units for therapy and rehabilitation," *Journal of Medical Imaging and Health Informatics*, vol. 9, no. 9, pp. 2031–2090, 2019.
- [9] L. V. Fulton, D. Dolezel, J. Harrop, Y. Yan, and C. P. Fulton, "Classification of Alzheimer's disease with and without imagery using gradient boosted machines and ResNet-50," *Brain Sciences*, vol. 9, no. 9, p. 212, 2019.
- [10] M. H. Guo and Y. Z. Du, "Classification of thyroid ultrasound standard plane images using ResNet-18 networks," in *2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, pp. 324–328, Xiamen, China, 2019.
- [11] J. P. Cohen, L. Dao, K. Roth et al., "Predicting COVID-19 pneumonia severity on chest X-ray with deep learning," *Cureus*, vol. 12, article e9448, 2020.
- [12] L. Li, L. Qin, Z. Xu 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, no. 2, pp. E65–E71, 2020.
- [13] X. G. Wang, X. B. Deng, Q. Fu et al., "A weakly-supervised framework for COVID-19 classification and lesion localization from chest CT," *IEEE Transactions on Medical Imaging*, vol. 39, no. 8, pp. 2615–2625, 2020.
- [14] Y. D. Zhang, S. C. Satapathy, L. Y. Zhu, J. M. Gorris, and S. H. Wang, "A seven-layer convolutional neural network for chest CT based COVID-19 diagnosis using stochastic pooling," *IEEE Sensors Journal*, 2020.
- [15] H. Ko, H. Chung, W. S. Kang 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, article e19569, p. 13, 2020.

- [16] S.-H. Wang, D. R. Nayak, D. S. Guttery, X. Zhang, and Y. D. Zhang, "COVID-19 classification by CCSHNet with deep fusion using transfer learning and discriminant correlation analysis," *Information Fusion*, vol. 68, pp. 131–148, 2021.
- [17] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," 2013, <https://arxiv.org/abs/1301.3557>.
- [18] Y.-D. Zhang, D. R. Nayak, X. Zhang, and S.-H. Wang, "Diagnosis of secondary pulmonary tuberculosis by an eight-layer improved convolutional neural network with stochastic pooling and hyperparameter optimization," *Journal of Ambient Intelligence and Humanized Computing*, 2020.
- [19] X. W. Jiang, M. Lu, and S. H. Wang, "An eight-layer convolutional neural network with stochastic pooling, batch normalization and dropout for fingerspelling recognition of Chinese sign language," *Multimedia Tools and Applications*, vol. 79, no. 21–22, pp. 15697–15715, 2020.
- [20] A. Jahanbakhshi, M. Momeny, M. Mahmoudi, and Y. D. Zhang, "Classification of sour lemons based on apparent defects using stochastic pooling mechanism in deep convolutional neural networks," *Scientia Horticulturae*, vol. 263, article 109133, p. 10, 2020.
- [21] B. C. Zhou, X. L. Wang, and Q. Q. Qi, "Optimal weights decoding of M-ary suprathreshold stochastic resonance in stochastic pooling network," *Chinese Journal of Physics*, vol. 56, no. 4, pp. 1718–1726, 2018.
- [22] H. Vu, H. C. Kim, M. Jung, and J. H. Lee, "fMRI volume classification using a 3D convolutional neural network robust to shifted and scaled neuronal activations," *Neuroimage*, vol. 223, article 117328, 2020.
- [23] J. Yang, Z. X. Ji, S. J. Niu, Q. Chen, S. T. Yuan, and W. Fan, "RMPPNet: residual multiple pyramid pooling network for subretinal fluid segmentation in SD-OCT images," *OSA Continuum*, vol. 3, no. 7, pp. 1751–1769, 2020.
- [24] B. Brzycki, A. P. Siemion, S. Croft et al., "Narrow-band signal localization for SETI on noisy synthetic spectrogram data," *Publications of the Astronomical Society of the Pacific*, vol. 132, article 114501, 2020.
- [25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, pp. 1–14, San Diego, CA, USA, 2015.
- [26] A. Hazra, P. Choudhary, S. Inunganbi, and M. Adhikari, "Bangla-Meitei Mayek scripts handwritten character recognition using convolutional neural network," *Applied Intelligence*, 2020.
- [27] Y. H. Sabry, W. Z. W. Hasan, A. H. Sabry, M. Z. A. A. Kadir, M. A. M. Radzi, and S. Shafie, "Measurement-based modeling of a semitransparent CdTe thin-film PV module based on a custom neural network," *IEEE Access*, vol. 6, pp. 34934–34947, 2018.
- [28] G. Kang, X. Dong, L. Zheng, and Y. Yang, "Patchshuffle regularization," 2017, <https://arxiv.org/abs/1707.07103>.
- [29] C. Shorten and T. M. Khoshgoftar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019.
- [30] S.-H. Wang, V. V. Govindaraj, J. M. Górriz, X. Zhang, and Y. D. Zhang, "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.
- [31] X. Li, Y. Chai, W. Chen, and F. Ao, "Identification of early esophageal cancer based on data augmentation," in *2020 39th Chinese Control Conference (CCC)*, pp. 6307–6312, Shenyang, China, 2020.
- [32] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, 2020.
- [33] J. Kim and J. M. Kim, "Bearing fault diagnosis using Grad-CAM and acoustic emission signals," *Applied Sciences*, vol. 10, no. 6, article 2050, 2020.
- [34] T. Akilan and Q. M. J. Wu, "sEnDec: an improved image to image CNN for foreground localization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4435–4443, 2020.
- [35] L. Guiga and A. W. Roscoe, "Neural network security: hiding CNN parameters with guided Grad-CAM," in *Proceedings of the 6th International Conference on Information Systems Security and Privacy (ICISSP)*, pp. 611–618, Valletta, Malta, 2020.
- [36] M. S. Hossain, G. Muhammad, and N. Guizani, "Explainable AI and mass surveillance system-based healthcare framework to combat COVID-19 like pandemics," *IEEE Network*, vol. 34, no. 4, pp. 126–132, 2020.
- [37] A. B. Tosun, F. Pullara, M. J. Becich, D. L. Taylor, J. L. Fine, and S. C. Chennubhotla, "Explainable AI (xAI) for anatomic pathology," *Advances in Anatomic Pathology*, vol. 27, no. 4, pp. 241–250, 2020.
- [38] M. S. Hossain, "Cloud-supported cyber-physical localization framework for patients monitoring," *IEEE Systems Journal*, vol. 11, no. 1, pp. 118–127, 2017.
- [39] M. S. Hossain and G. Muhammad, "Emotion-aware connected healthcare big data towards 5G," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2399–2406, 2018.
- [40] S. Liu, D. Y. Liu, G. Srivastava, D. Połap, and M. Woźniak, "Overview and methods of correlation filter algorithms in object tracking," *Complex & Intelligent Systems*, vol. 23, 2020.
- [41] S. Liu, W. L. Bai, G. Srivastava, and J. A. T. Machado, "Property of self-similarity between baseband and modulated signals," *Mobile Networks and Applications*, vol. 25, no. 4, pp. 1537–1547, 2020.
- [42] S. Liu, X. Y. Liu, S. Wang, and K. Muhammad, "Fuzzy-aided solution for out-of-view challenge in visual tracking under IoT-assisted complex environment," *Neural Computing & Applications*, vol. 33, no. 4, pp. 1055–1065, 2021.
- [43] S. Liu, S. Wang, X. Liu, C. Lin, and Z. Lv, "Fuzzy detection aided real-time and robust visual tracking under complex environments," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 1, 2020.
- [44] S. Liu, C. L. Guo, F. al-Turjman, K. Muhammad, and V. H. C. de Albuquerque, "Reliability of response region: a novel mechanism in visual tracking by edge computing for IIoT environments," *Mechanical Systems and Signal Processing*, vol. 138, article 106537, 2020.
- [45] X. Mao, L. Ding, Y. Zhang, R. Zhan, and S. Li, "Knowledge-aided 2-D autofocus for spotlight SAR filtered backprojection imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 9041–9058, 2019.
- [46] X. H. Mao, X. L. He, and D. Q. Li, "Knowledge-aided 2-D autofocus for spotlight SAR range migration algorithm

- imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 9, pp. 5458–5470, 2018.
- [47] X. H. Mao and D. Y. Zhu, “Two-dimensional autofocus for spotlight SAR polar format imagery,” *IEEE Transactions on Computational Imaging*, vol. 2, pp. 524–539, 2016.
- [48] K. Xia, Y. Zhang, Y. Jiang et al., “TSK fuzzy system for multi-view data discovery underlying label relaxation and cross-rule & cross-view sparsity regularizations,” *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020.
- [49] K. Xia, X. Gu, and B. Chen, “Cross-dataset transfer driver expression recognition via global discriminative and local structure knowledge exploitation in shared projection subspace,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2020.
- [50] K. Xia, T. Ni, H. Yin, and B. Chen, “Cross-domain classification model with knowledge utilization maximization for recognition of epileptic EEG signals,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, pp. 53–61, 2020.