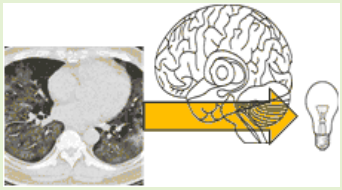
A seven-layer convolutional neural network for chest CT based COVID-19 diagnosis using stochastic pooling

Yu-Dong Zhang, *SMIEEE*, Suresh Chandra Satapathy, *SMIEEE*, Li-Yao Zhu, Juan Manuel Górriz, Shui-Hua Wang, *SMIEEE*,

***Abstract*—(Aim) COVID-19 pandemic causes numerous death tolls till now. Chest CT is an effective imaging sensor system to make accurate diagnosis. (Method) This paper proposed a novel seven layer convolutional neural network based smart diagnosis model for COVID-19 diagnosis (7L-CNN-CD). We proposed a 14-way data augmentation to enhance the training set, and introduced stochastic pooling to replace traditional pooling methods. (Results) The 10 runs of 10-fold cross validation experiment show that our 7L-CNN-CD approach achieves a sensitivity of 94.44±0.73, a specificity of 93.63±1.60, and an accuracy of 94.03±0.80. (Conclusion) Our proposed 7L-CNN-CD is effective in diagnosing COVID-19 in chest CT images. It gives better performance than several state-of-the-art algorithms. The data augmentation and stochastic pooling methods are proven to be effective.**



***Index Terms*—deep learning; convolutional neural network; data augmentation; stochastic pooling; COVID-19**

1 Introduction

COVID-19 (also known as coronavirus) was declared a Public Health Emergency of International Concern on 30/01/2020, and declared as a pandemic on 11/03/2020.

Till 2/Sep, this COVID-19 pandemic caused 25.8 million confirmed cases and 858.2 thousand death tolls (US 187.4k deaths, Brazil 122.5k deaths, India 66.3k deaths, Mexico 65.2k deaths, UK 41.5k deaths, etc.)

Global economy experienced negative effects from COVID-19. For example, Balsalobre-Lorente, et al. [1] analyzed consequences of COVID-19 on the social isolation of Chinese economy. Chaudhary, et al. [2] presented reflections for policy and program of the effect of COVID-19 on economy in India.

Two prevail diagnosis are available. One is viral testing via a nasopharyngeal swab to test the presence of viral RNA fragments [3]. Another is imaging methods, among which the chest computed tomography (CCT) [4] is one of the imaging devices that can provide the highest sensitivity. The CCT uses X-ray generator and X-ray sensors that rotate around the subjects.

The main biomarkers in CCT differentiating COVID-19 from healthy people are the asymmetric peripheral ground-glass opacities (GGOs) without pleural effusions [5]. This study collects those CCT slices.

However, manual interpretation by radiologists is tedious and easy to be influenced by inter-expert and intra-expert factors (such as fatigue, emotion, etc.). Smart diagnosis systems via computer vision and artificial intelligence can benefit patients, radiologists, experts and hospitals. Traditional artificial intelligence (AI) and modern deep learning (DL) methods have achieved excellent results in analyzing medical images, e.g., Lu [6] proposed a radial-basis-function neural network (RBFNN) to detect pathological brains. Yang [7] presented a kernel-based extreme learning classifier (K-ELM) to create a novel pathological brain detection system. Their method was robust and effective. Lu [8] proposed a novel extreme learning machine trained by the bat algorithm (ELM-BA) approach. Li and Liu [9] introduced the real-coded biogeography-based optimization (RCBBO) to detect diseased brains. Jiang [10] used a six-layer convolutional neural network (6L-CNN) to recognize sign language fingerspelling. Szegedy, et al. [11] presented the GoogleNet. Yu and Wang [12] suggested the use of ResNet18 for mammogram abnormality detection. Furthermore, some smart health systems gained success in emotion-aware security [13], authentication [14], and IoT [15].

We proposed a novel 7-layer convolutional neural network for COVID-19 diagnosis (7L-CNN-CD). To improve its performance, three improvements were proposed in this study: (i) A 12-way data augmentation (DA-12) was proposed; (ii) Stochastic pooling was introduced to replace traditional pooling methods;

2 Dataset

Image acquisition CT configuration and method: Philips Ingenuity 64 row spiral CT machine, KV: 120, MAS: 240, layer thickness 3 mm, layer spacing 3 mm, screw pitch 1.5: lung window (W: 1500 HU, L: -500 HU) , Mediastinum window (W: 350 HU, L: 60 HU), thin layer reconstruction according to the lesion display, layer thickness and layer distance are 1mm lung window image. The patients were placed in a supine position,

This paper is partially supported by Natural Science Foundation of China (61602250); Henan Key Research and Development Project (182102310629); Fundamental Research Funds for the Central Universities (CDLS-2020-03); Key Laboratory of Child Development and Learning Science (Southeast University), Ministry of Education; 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).

YDZ & SCS contributed equally and should be regarded as co-first authors.
Co-correspondence should be addressed to SHW, LYZ, JMG
YDZ is with School of Informatics, University of Leicester, Leicester, LE1 7RH, UK (e-mail: yudongzhang@ieee.org)
SCS is with School of Computer Engg, KIIT Deemed to University, Bhubaneswar, India (e-mail: sureshsatapathy@ieee.org)
LYZ is with Department of Infection, Huai'an Fourth People's Hospital, Huai'an, China (e-mail: zhu_liyao@126.com)
JMG is with Department of Signal Theory, Networking and Communications, University of Granada, Granada, Spain (e-mail: gorriz@ugr.es)
SHW is with School of Architecture Building and Civil engineering, Loughborough University, Loughborough, LE11 3TU, UK (e-mail: shuihuawang@ieee.org)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3025855, IEEE Sensors Journal

breathing deeply after holding in, and conventionally scanned from the lung tip to the costal diaphragm angle.

For each subject, 1-4 slices were chosen. Slice level selection (SLS) method was employed: For COVID-19 pneumonia patients, the slice showing the largest size and number of lesions was selected. For normal subjects, any level of the image can be selected. The resolutions of all images are $1{,}024 \times 1{,}024$. Table 1 shows the demographics, where HC means healthy control.

Table 1 demographics of subjects used in this study

	No. of subjects (m/f)	No. of Images	Age Range
COVID-19	142 (95/47)	320	22-91
HC	142 (88/54)	320	21-76

When there are differences between the two analyses $(\mathcal{J}_1, \mathcal{J}_2)$, a superior doctor $(\mathcal{S})$ was consulted to reach a consensus. Suppose X means a CCT image scan, $\mathcal{M}$ means the labelling of each individual expert, and the final labelling $\bar{\mathcal{M}}$ is obtained by

$$\bar{\mathcal{M}}(X) = \begin{cases} \mathcal{M}(\mathcal{J}_1) & \mathcal{M}(\mathcal{J}_1) = \mathcal{M}(\mathcal{J}_2) \\ \mathrm{MV}(\mathcal{M}_{all}) & \text{otherwise} \end{cases} \quad (1.a)$$

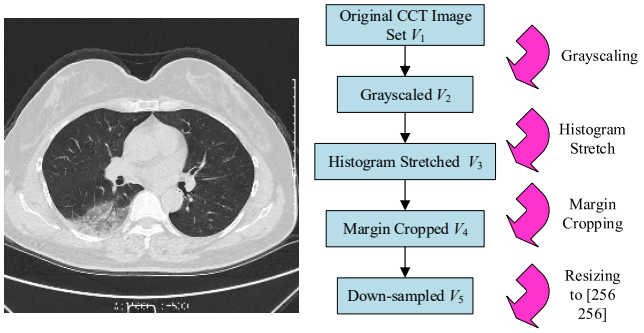
$$\mathcal{M}_{all} = [\mathcal{M}(\mathcal{J}_1), \mathcal{M}(\mathcal{J}_2), \mathcal{M}(\mathcal{S})] \quad (1.b)$$

where MV denotes majority voting, $\mathcal{M}_{all}$ represents the labelling of all three experts.

3 Methodology

Table 8 shows the abbreviations and their full names for ease of understanding of our methodology part.

3.1 Preprocessing



(a) A raw COVID-19 image (b) Preprocessing

Figure 1 Preprocessing on raw dataset

The original dataset containing 320 COVID-19 images and 320 HC images is symbolized as V_1, each image is symbolized as $v_1(i) \in V_1, i = 1,2,\cdots,n = 640$. We have

$$V_1 = \{v_1(1), v_1(2), \cdots, v_1(i), \cdots, v_1(640)\} \quad (2)$$

Figure 1(a) shows a raw COVID-19 CCT image. Figure 1(b) shows the flowchart of our preprocessing procedure. First, we converted all color images to grayscale by only reserving the luminance information. The reason of performing grayscale is there is no need to store a grayscale image in three color channels. Directly inputting original RGB images to the neural network will increase the computation burdens. Thus, we get the grayscale image set V_2 as

$$\begin{aligned} V_2 &= \mathcal{G}(V_1|\mathrm{RGB} \to \mathrm{Grayscale}) \\ &= \{v_2(1), v_2(2), \ldots, v_2(i), \ldots v_2(640)\} \end{aligned} \quad (3)$$

where $\mathcal{G}$ means the grayscale operation.

Second, histogram stretching (HS) method was used to increase every slice's contrast. For i-th image $v_2(i), i = 1,2,\cdots,640$, we first calculate their minimum grayscale value $\mu_{min}[v_2(i)]$ and maximum grayscale value $\mu_{max}[v_2(i)]$ respectively by

$$\mu_{min}[v_2(i)] = \min_{x,y=1}^{1024} v_2(i|x,y) \quad (4.a)$$

$$\mu_{max}[v_2(i)] = \max_{x,y=1}^{1024} v_2(i|x,y) \quad (4.b)$$

here (x, y) means coordinates of pixel of the image $v_2(i)$. The new histogram stretched image $v_3(i)$ is obtained by

$$v_3(i) = \frac{v_2(i) - \mu_{min}[v_2(i)]}{\mu_{max}[v_2(i)] - \mu_{min}[v_2(i)]} \quad (5.a)$$

$$\begin{aligned} V_3 &= HS(V_2) \\ &= \{v_3(1), v_3(2), \cdots, v_3(i), \ldots V_3(640)\} \end{aligned} \quad (5.b)$$

In all, we get the histogram stretched image set V_3 as above.

Third, we crop the images to remove the texts at the margin areas, and the checkup bed at the bottom area. Thus, we get the cropped dataset V_4 as

$$\begin{aligned} V_4 &= C(V_3, [\text{top, bottom, left, right}]) \\ &= \{v_4(1), v_4(2), \cdots, v_4(i), \cdots, v_4(640)\} \end{aligned} \quad (6.a)$$

$$\text{top} = \text{bottom} = \text{left} = \text{right} = 150 \quad (6.b)$$

where C represents crop operation. Four crop variables: top, bottom, left, and right means the pixels to be removed during crop operation. In this study all their values equal 150. Now the size of each image is reduced from 1024×1024 to 724×724.

Fourth, we downsampled each image to size of [256, 256], and we now get the resized image set V_5 as

$$\begin{aligned} V_5 &= \Downarrow (V_4, [256\ 256]) \\ &= \{v_5(1), v_5(2), \ldots, v_5(i), \ldots v_5(640)\} \end{aligned} \quad (7)$$

where $\Downarrow: x \mapsto y$ means the downsampling (DS) function, where y is a downsampled image of original image x.

Table 2 compares the size and storage of each image $v_s(i), s = 1,\cdots,5, i = 1,\cdots,640$ at every preprocessing step. We can see here after preprocessing procedure, each image will only cost about 2.08% of its original storage or size. The compression ratio (CR) rates of i-th image of final state V_5 to original stage V_1 were calculated by following equation.

$$\mathrm{CR}_{\mathrm{Storage}}(i) = \frac{\mathrm{byte}(v_5(i))}{\mathrm{byte}(v_1(i))} = \frac{262{,}144}{12{,}582{,}912} = 2.083\% \quad (8.a)$$

$$\mathrm{CR}_{\mathrm{size}}(i) = \frac{\mathrm{size}(v_5(i))}{\mathrm{size}(v_1(i))} = \frac{65{,}536}{3{,}145{,}728} = 2.083\% \quad (8.b)$$

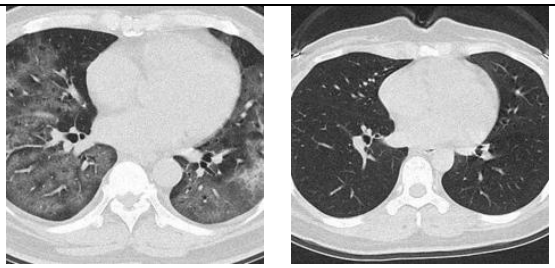
$$\forall i \in [1{,}640], \mathrm{CR}_{\mathrm{size}}(i) = \mathrm{CR}_{\mathrm{size}}(i) \quad (8.c)$$

We can see here the storage CR equals size CR for any i-th image. Figure 2 shows two samples from the preprocessed dataset V_5.

Table 2 Image size and storage per image at each preprocessing step

Preprocess	Symbol	Size (per image)	Storage (per image)
Original	$v_1(i)$	$1024 \times 1024 \times 3 = 3{,}145{,}728$	12,582,912
Grayscale	$v_2(i)$	$1024 \times 1024 \times 1 = 1{,}048{,}576$	4,194,304
HS	$v_3(i)$	$1024 \times 1024 \times 1 = 1{,}048{,}576$	4,194,304
Crop	$v_4(i)$	$724 \times 724 \times 1 = 524{,}176$	2,096,704
DS	$v_5(i)$	$256 \times 256 \times 1 = 65{,}536$	262,144

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3025855, IEEE Sensors Journal



(a) COVID-19 (b) HC

Figure 2 Two samples of preprocessed dataset V_5

3.2 Improvement I: Data Augmentation

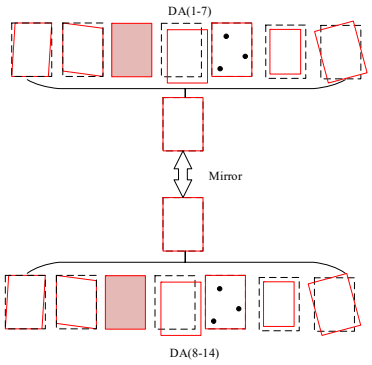


Figure 3 Illustration of our DA-14

Generally, the CCT image set faces small-size dataset (SSD) and lack of generalization (LoG) problems. To break the curse of SSD and LoG, there are four possible types of solutions: (i) data generation (DG); (ii) regularization approach (RA), (iii) ensemble approach (EA); and (iv) data augmentation (DA). All those DG, RA, EA, and DA methods are effective in handling SSD and LoG problems.

We proposed a 14-way DA method, as shown in Figure 3. We will use 10-fold cross validation technique. Suppose the preprocessed CCT image set V_5 will split into ten folds, nine of which form the training set B, and the rest forms test set C.

$$V_5:\begin{cases}\xrightarrow{\text{1st trial}}\{B_1, C_1\}\\ \cdots\\ \xrightarrow{\text{10th trial}}\{B_{10}, C_{10}\}\end{cases} \tag{9.a}$$

$$\|B_r + C_r\| = \|V_5\|, \forall r \in (1{,}10) \tag{9.b}$$

where $\|x\|$ means the cardinality of the set x. For ease of reading, we ignore the run-index r in following texts, and just simplify the situations as $V_5 \xrightarrow{\text{split}} \{B, C\}$, and we assume B contains $\|B\|$ images

$$b(k) \in B, k = 1, \cdots, \|B\| \tag{10}$$

For each image $b(k)$, we shall define all the 14 different DA operations.

(i) Rotation. Rotation angle $\overrightarrow{\gamma^{rot}}$ was in the value from -30° to 30° in increase of 2°, skipping the value of $\gamma^{rot} = 0$, since it corresponds to the original image $b(k)$.

$$\begin{aligned}\overrightarrow{b^1(k)} &= \text{rotate}\left[b(k), \overrightarrow{\gamma^{rot}}\right]\\ &= [b_1^{rot}(k, \gamma_1^{rot}), b_2^{rot}(k, \gamma_2^{rot}), \cdots, b_{30}^{rot}(k, \gamma_{30}^{rot})]\end{aligned} \tag{11}$$

where the rotation factor vector $\overrightarrow{\gamma^{rot}}$ is defined as

$$\gamma_1^{rot} = -30°, \gamma_2^{rot} = -28°, \cdots, \gamma_{15}^{rot} = -2° \tag{12.a}$$

$$\gamma_{16}^{rot} = 2°, \gamma_{17}^{rot} = 4° \cdots, \gamma_{30}^{rot} = 30° \tag{12.b}$$

(ii) Scaling. All training CCT images were scaled with scaling factor $\overrightarrow{\gamma^{scale}}$, the values of which vary from 0.7 to 1.3 with increase of 0.02, skipping the value of 1.

$$\begin{aligned}\overrightarrow{b^2(k)} &= \text{scale}\left[b(k), \overrightarrow{\gamma^{scale}}\right]\\ &= [b_1^{scale}(k, \gamma_1^{scale}), \ldots b_{30}^{scale}(k, \gamma_{30}^{scale})]\end{aligned} \tag{13}$$

Where scaling factor vector $\overrightarrow{\gamma^{scale}}$ is defined as

$$\gamma_1^{scale} = 0.7, \gamma_2^{scale} = 0.72, \ldots, \gamma_{15}^{scale} = 0.98 \tag{14.a}$$

$$\gamma_{16}^{scale} = 1.02, \gamma_{17}^{scale} = 1.04, \ldots, \gamma_{30}^{scale} = 1.3 \tag{14.b}$$

(iii) Noise injection (NI). The m-mean v-variance Gaussian noises were added to the all CCT training images to produce 30 new noised images.

$$\begin{aligned}\overrightarrow{b^3(k)} &= \text{NI}\left[a(k), \overrightarrow{m^{NI}}, \overrightarrow{v^{NI}}\right]\\ &= \left[b_1^{NI}(k, m_1^{NI}, v_1^{NI}), \ldots b_{30}^{noise}(k, m_{30}^{NI}, v_{30}^{NI})\right]\end{aligned} \tag{15}$$

where the mean and variance vector definition of noise are defined as $m_1^{NI} = m_2^{NI} = \cdots = m_{30}^{NI} = 0$, $v_1^{NI} = v_2^{NI} = \cdots = v_{30}^{NI} = 0.01$. The values of 0 and 0.01 are default values of mean and variance of Gaussian noises, respectively.

(iv) Random translation (RT). All CCT image $b(k)$ was translated 30 times with random horizontal shift vector $\overrightarrow{\gamma^{xs}}$ and random vertical shift $\overrightarrow{\gamma^{ys}}$

$$\begin{aligned}\overrightarrow{b^4(k)} &= \text{RT}\left[b(k), \overrightarrow{\gamma^{xs}}, \overrightarrow{\gamma^{ys}}\right]\\ &= \left[b_1^{RT}\left(k, \gamma_1^{xs}, \gamma_1^{ys}\right), \ldots b_{30}^{RT}\left(k, \gamma_{30}^{xs}, \gamma_{30}^{ys}\right)\right]\end{aligned} \tag{16}$$

where the values of $\overrightarrow{\gamma^{xs}}$ and $\overrightarrow{\gamma^{ys}}$ are in the range of [-15, 15], and obey uniform distribution $\mathbb{N}$.

$$\forall j \in (1{,}30), \begin{cases}\gamma_j^{xs} \sim \mathbb{N}[-15{,}15]\\ \gamma_j^{ys} \sim \mathbb{N}[-15{,}15]\end{cases} \tag{17}$$

(v) Gamma correction (GC). GC can help adjust the contrast of original image [16]. The factor vector of GC $\overrightarrow{\gamma^{GC}}$ varied from 0.4 to 1.6 with increase of 0.04, skipping the value of 1.

$$\begin{aligned}\overrightarrow{b^5(k)} &= \text{GC}\left[b(k), \overrightarrow{\gamma^{GC}}\right]\\ &= [b_1^{GC}(k, \gamma_1^{GC}), \ldots b_{30}^{GC}(k, \gamma_{30}^{GC})]\end{aligned} \tag{18}$$

where the values of $\overrightarrow{\gamma^{GC}}$ is chosen as:

$$\gamma_1^{GC} = 0.4, \gamma_2^{GC} = 0.44, \cdots, \gamma_{15}^{GC} = 0.96 \tag{19.a}$$

$$\gamma_{16}^{GC} = 1.04, \gamma_{17}^{GC} = 1.08, \cdots, \gamma_{30}^{GC} = 1.6 \tag{19.b}$$

(vi) Horizontal shear transform (HST). We will generate 30 horizontal shear transform (HST) images as

$$\begin{aligned}\overrightarrow{b^6(k)} &= \text{HST}\left[b(k), \overrightarrow{\gamma^{HST}}\right]\\ &= [b_1^{HST}(k, \gamma_1^{HST}), \ldots b_{30}^{HST}(k, \gamma_{30}^{HST})]\end{aligned} \tag{20}$$

where the HST values are assigned from -0.15 to 0.15 with increase of 0.01, skipping the value o 0

$$\gamma_1^{HST} = -0.15, \gamma_2^{HST} = -0.14, \ldots, \gamma_{15}^{HST} = -0.01 \tag{21.a}$$

$$\gamma_{16}^{HST} = 0.01, \gamma_{17}^{HST} = 0.02, \ldots, \gamma_{30}^{HST} = 0.15 \tag{21.b}$$

(vii) Vertical shear transform (VST). Similarly, we generate 30 vertical shear transform (VST) images as below. Besides, the values of VST factor vector $\overrightarrow{\gamma^{VST}}$ are the same as $\overrightarrow{\gamma^{HST}}$.

$$\begin{aligned}\overrightarrow{b^7(k)} &= \text{VST}\left[b(k), \overrightarrow{\gamma^{VST}}\right]\\ &= [b_1^{VST}(k, \gamma_1^{VST}), \ldots b_{30}^{VST}(k, \gamma_{30}^{VST})]\end{aligned} \tag{22.a}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3025855, IEEE Sensors Journal

$$\gamma_j^{VST} = \gamma_j^{HST}, \forall j \in [1,30] \quad (22.b)$$

(viii) Mirror. The original image $b(k)$ is mirrored and we obtain a new image $b'(k)$. Suppose M is the mirror function, we have

$$b'(k) = M[b(k)] \quad (23)$$

we define following operations:

$$\begin{cases} \overrightarrow{b^8(k)} = M\left[\overrightarrow{b^1(k)}\right] \\ \overrightarrow{b^9(k)} = M\left[\overrightarrow{b^2(k)}\right] \\ \cdots \\ \overrightarrow{b^{14}(k)} = M\left[\overrightarrow{b^7(k)}\right] \end{cases} \quad (24)$$

(ix) Concatenation. All the first seven DA results are concatenated, and we have

$$\underbrace{\overrightarrow{b^{DA(1-7)}(k)}}_{210} = \mathbb{C}\left(\underbrace{\overrightarrow{b^1(k)}}_{30}, \underbrace{\overrightarrow{b^2(k)}}_{30}, \cdots, \underbrace{\overrightarrow{b^6(k)}}_{30}, \underbrace{\overrightarrow{b^7(k)}}_{30}\right) \quad (25)$$

where $\mathbb{C}$ means the concatenation. The size of $\overrightarrow{b^{DA(1-7)}(k)}$ is $30 \times 7 = 210$ images, then we have the results of 8-14 DA techniques as

$$\overrightarrow{b^{DA(8-14)}(k)} = \mathbb{C}_{j=8}^{14} \overrightarrow{b^j(k)} \quad (26)$$

Finally, one original image $b(k)$ will yield to 365 images (containing itself) in the enhanced training set.

$$\underbrace{\overrightarrow{b^{DA}(k)}}_{421} \xrightarrow{DA} \mathbb{C}\left(\underbrace{b(k)}_{1}, \underbrace{\overrightarrow{b^{DA(1-7)}(k)}}_{210}, \underbrace{\overrightarrow{b^{DA(8-14)}(k)}}_{210}\right) \quad (27)$$

3.3 Improvement 2: Stochastic Pooling

In traditional CNN, the activation maps (AMs) are usually too large [17] (i.e., contain too many features) which will cause (i) overfitting of the training and (ii) large computational costs. Thus, pooling layers (PLs) are frequently used to reduce the size of AMs. Besides, PL could help guarantee the characteristics of invariance-to-translation. There exist three generally-used pooling techniques: (i) l_2 norm pooling (L2P); (ii) average pooling (AP); and (iii) max pooling (MP). Assume pooling is a function $P: \mathcal{S} \mapsto t$.

L2P calculates the l_2 norm [18] of a given region $\mathcal{S}$. Suppose

$$\mathcal{S} = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \quad (28)$$

L2P output t^{L2P} is defined as $t^{L2P}(\mathcal{S}) = sqrt\left(\sum_{i,j=1}^{2} s_{ij}^2\right)$. In this study, we add a constant 1/4 under the square root to make it easier to compare with other pooling methods. This constant 1/4 does not influence training and inference.

$$t^{L2P}(\mathcal{S}) = \sqrt{\frac{\sum_{i,j=1}^{2} s_{ij}^2}{4}} \quad (29)$$

The AP [19] calculates the mean value of region $\mathcal{S}$

$$t^{AP}(\mathcal{S}) = \frac{\sum_{i,j=1}^{2} s_{ij}}{4} \quad (30)$$

Finally, MP picks out the maximal value from region $\mathcal{S}$

$$t^{MP}(\mathcal{S}) = \max_{i,j=1}^{2} s_{ij} \quad (31)$$

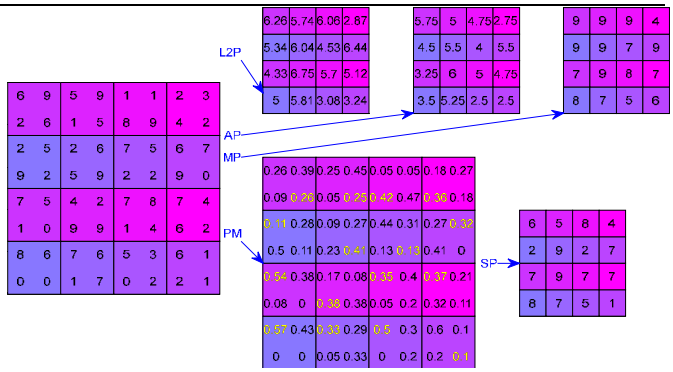


Figure 4 A toy example of four pooling techniques
(L2P = l_2 norm pooling; AP = average pooling; MP =max pooling; SP = stochastic pooling)

Figure 4 showcases the differences of our pooling methods, where we assume both pooling size and pooling stride equal 2. Observe the top left region $\bar{\mathcal{S}}$, its vectorization is $\text{vec}(\bar{\mathcal{S}}) = [6 \quad 9 \quad 2 \quad 6]$. The calculation of L2P, AP, and MP are as below: $t^{L2P}(\bar{\mathcal{S}}) = \text{sqrt}\left(\frac{6^2+9^2+2^2+6^2}{4}\right) = \text{sqrt}\left(\frac{157}{4}\right) = 6.26$, $t^{AP}(\bar{\mathcal{S}}) = \frac{6+9+2+6}{4} = 5.75$, $t^{SP}(\bar{\mathcal{S}}) = \max(6,9,2,6) = 9$.

The SP was invented to conquer the problems caused by aforementioned three pooling methods: L2P, MP and AP. Both L2P and AP does not work well, since all pixels in $\mathcal{S}$ are considered by L2P and AP, thus they could reduce the values of strong activations because of other surrounding near-zero pixels. On the other hand, the MP elucidates this obstruction, although it simply overfits the training set and causes the LoG problem.

Instead of computing the l_2 norm, average value or max value, the output of the SP t^{SP} is attained via sampling from a multinomial distribution [20] formed from the activations of each element in region $\mathcal{S}$ [21].

(1) Reckon the probability p_{ij} of each element $\{s_{ij}, \forall i,j = 1,2\} \in \mathcal{S}$.

$$p_{ij} = \frac{s_{i,j}}{\sum_{i,j=1}^{2} s_{ij}} \quad (32.a)$$

$$\sum_{i,j=1}^{2} p_{i,j} = 1 \quad (32.b)$$

(2) Select a location α within the $\mathcal{S}$ in accordance with the probability $\{p_{ij}\} \in \mathcal{P}$, calculated by scanning the $\mathcal{S}$ from up to bottom and left to right [22].

$$\alpha \sim (p_{11}, p_{12}, p_{21}, p_{22}) \quad (33)$$

(3) The output is the value at location α.

$$t^{SP}(\mathcal{S}) = s_\alpha \quad (34)$$

We use the first block $\bar{\mathcal{S}}$ in Figure 4 as an instance. The calculation procedures of SP are described below:

$$\mathcal{P}(\bar{\mathcal{S}}) = \begin{bmatrix} 6 & 9 \\ 2 & 6 \end{bmatrix} / \sum \left(\begin{bmatrix} 6 & 9 \\ 2 & 6 \end{bmatrix}\right) = \begin{bmatrix} 0.26 & 0.39 \\ 0.09 & 0.26 \end{bmatrix} \quad (35)$$

Thus, we get $\alpha(\bar{\mathcal{S}}) = (2 \quad 2)$, and $t^{SP}(\bar{\mathcal{S}}) = 6$. Using the probability map $\mathcal{P}(\bar{\mathcal{S}})$, we randomly select the position $\alpha = (2 \quad 2)$ associated with probability of $p_{22} = 0.26$. Thus, the output $t^{SP}(\mathcal{S})$ of SP at region $\bar{\mathcal{S}}$ is 6. Instead of considering the max values barely or considering all the elements in the region, SP uses non-maximal activations randomly within the region $\mathcal{S}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3025855, IEEE Sensors Journal

3.4 Measures and Indicators

We set a 10-fold cross validation on the whole dataset V_5. Each fold will contain 32 COVID-19 images and 32 HC images. Within each trial, the training set contains $288 + 288 = 576$ images, and the test set contains $32 + 32 = 64$ images. After combining all the 10 trials, the test set will contain 640 images. The above 10-fold cross validation will run 10 times, and so the final report was based on $10 \times 640 = 6{,}400$ images. Table 3 shows the split setting of our dataset.

Table 3 Split setting of our dataset

Set	Percentage	COVID-19	HC	Total
Training B_t	90%	288	288	576
DA Training		121,248	121,248	242,496
Test C_t	10%	32	32	64
Total	100%	320	320	640

This proposed seven-layer convolutional neural network for COVID-19 diagnosis (7L-CNN-CD) will be tested by 10 runs of 10-fold cross validation. Suppose the ideal confusion matrix D over the test set at t-th trial and r-th run is

$$D^{ideal}(t,r) = \begin{bmatrix} 32 & 0 \\ 0 & 32 \end{bmatrix} \tag{36}$$

Where the value 32 can be found in the test row in Table 3. The value of 32 means the number of COVID-19 cases and the number of HC cases in the test set. After running through 1-10 trials, and we get the confusion matrix of one-run 10-fold CV as

$$D^{ideal}(r) = \sum_{t=1}^{10} D^{ideal}(t,r) = \begin{bmatrix} 320 & 0 \\ 0 & 320 \end{bmatrix} \tag{37}$$

In realistic inference, we cannot get the perfect diagonal matrix, where all off-diagonal elements are zero. Suppose the confusion matrix at r-th run is

$$D^{real}(r) = \sum_{t=1}^{10} D^{real}(t,r) = \begin{bmatrix} d_{11}(r) & d_{12}(r) \\ d_{21}(r) & d_{22}(r) \end{bmatrix} \tag{38}$$

Note $0 \le d_{ij} \le 320, \forall i,j = 1,2$ in this study. Here d_{11} and d_{22} represent true positive (TP) and true negative (TN), respectively. Positive class (P) is COVID-19, and negative class (N) is healthy control. d_{12} and d_{21} represent false negative (FN) and false positive (FP), respectively. We can define four simple measures as

$$\beta^1(r) = \frac{d_{11}(r)}{d_{11}(r)+d_{12}(r)} \tag{39.a}$$

$$\beta^2(r) = \frac{d_{22}(r)}{d_{21}(r)+d_{22}(r)} \tag{39.b}$$

$$\beta^3(r) = \frac{d_{11}(r)}{d_{11}(r)+d_{21}(r)} \tag{39.c}$$

$$\beta^4(r) = \frac{d_{11}(r)+d_{22}(r)}{\eta(r)} \tag{39.d}$$

$$\eta(r) = d_{11}(r) + d_{12}(r) + d_{21}(r) + d_{22}(r) \tag{39.e}$$

Three advanced measures are defined below. F1 score is:

$$\beta^5(r) = 2 \times \frac{\beta^3(r) \times \beta^1(r)}{\beta^3(r) + \beta^1(r)} = \frac{2 \times d_{11}(r)}{2 \times d_{11}(r) + d_{21}(r) + d_{12}(r)} \tag{40}$$

Matthews correlation coefficient (MCC) is defined as

$$\beta^6(r) = \frac{d_{22}(r) \times d_{11}(r) - d_{21}(r) \times d_{12}(r)}{\sqrt{\theta(r)}} \tag{41.a}$$

$$\theta(r) = [d_{21}(r) + d_{11}(r)] \times [d_{11}(r) + d_{12}(r)] \times [d_{22}(r) + d_{21}(r)] \times [d_{22}(r) + d_{12}(r)] \tag{41.b}$$

Fowlkes–Mallows index (FMI) is defined as

$$\beta^7(r) = \sqrt{\frac{d_{11}(r)}{d_{11}(r)+d_{21}(r)} \times \frac{d_{11}(r)}{d_{11}(r)+d_{12}(r)}} \tag{42}$$

After combining 10 runs $r \in [1,10]$, we can calculate the mean and standard deviation (SD) of all m-th $(\forall m \in [1,7])$ measures as

$$\text{mean}(\beta^m) = \frac{1}{10} \times \sum_{r=1}^{10} \beta^m(r) \tag{43.a}$$

$$\text{std}(\beta^m) = \sqrt{\frac{1}{9} \times \sum_{r=1}^{10} [\beta^m(r) - \text{mean}(\beta^m)]^2} \tag{43.b}$$

3.5 Proposed 7L-CNN-CD Algorithm

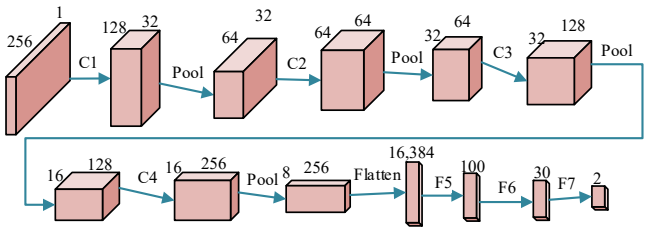


Figure 5 Structure of proposed 7-layer CNN

Figure 5 presents the structure of proposed 7-layer CNN (7L-CNN). After training, the network can be used to diagnose COVID-19 is called 7L-CNN-CD. The sizes of activation map are labelled at each cube in Figure 5. Table 4 shows the pseudocode of our 7L-CNN-CD model. Here we divide our algorithm into two phases: (I) Preprocessing and (II) 10 runs of 10-fold cross validation.

Table 4 Pseudocode of our 7L-CNN-CD model

Input: Original Image Set V_1
Ground Truth: $\mathcal{M}$ obtained from two junior and one senior radiologists. See Eq. (1.a)

Phase I: Preprocessing

Grayscale $V_1 \rightarrow V_2$. See Eq. (3)
Histogram Stretching $V_2 \rightarrow V_3$. See Eq. (5.a)
Image Crop $V_3 \rightarrow V_4$. See Eq. (6.a)
Downsampling $V_4 \rightarrow V_5$. See Eq. (7)

Phase II: 10 runs of 10-fold cross validation

for $r = 1{:}10$ % r is run index
  Randomly split preprocessed set V_5 into 10 folds
  $V_5 \xrightarrow{\text{split}} \{V_5^r(1), V_5^r(2), \cdots, V_5^r(10)\}$,
  for $t = 1{:}10$ % t is trial index
    Step II.A: Training & Test Set
      Test Set. C is chosen as the t-th fold.
      $C(r,t) = V_5^r(t)$;
      Training Set. B is chosen as the other folds.
      $B(r,t) = \{V_5^r(1), \cdots, V_5^r(t-1), V_5^r(t+1), \cdots, V_5^r(10)\}$.
      Enhanced Training Set.
      DA$[B(r,t)]$, see equation (27).
    Step II.B: Create Initial CNN model
      Create an initial deep network $\mathbb{E}(r,t)$ via 7L-CNN model;
      Use SP to replace all pooling layers in 7L-CNN model. See equation (34).
    Step II.C Trained 7L-CNN-CD model
      Train 7L-CNN network using DA$[B(r,t)]$ and ground truth $\mathcal{M}$
      Trained model $\breve{\mathbb{E}}(r,t)$:
      $\breve{\mathbb{E}}(r,t) = \text{trainnetwork}\{\mathbb{E}(r,t), \text{DA}[B(r,t)], \mathcal{M}\}$;
    Step II.D: Confusion Matrix Performance

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3025855, IEEE Sensors Journal

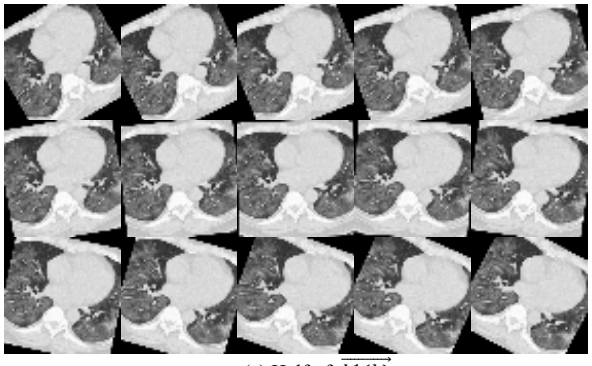
Test prediction $Pred(r,t)$:
$Pred(r,t) = \text{predict}\left[\tilde{\mathbb{E}}(r,t), C(r,t)\right]$;
Test performance. $D^{real}(r,t)$ is obtained by comparing test prediction and ground truth.
$D^{real}(r,t) = \text{compare}[\mathcal{M}, Pred(r,t)]$.
end
Summarize all 10 trials and get $D^{real}(r)$, see Eq. (38).
Calculate $\beta^m(r), m = 1,2,\cdots,7$, see Eqs. (39.a)-(42)
end
Output mean and SD of β^m. see Eq. (43.a)

4 Results, and discussions

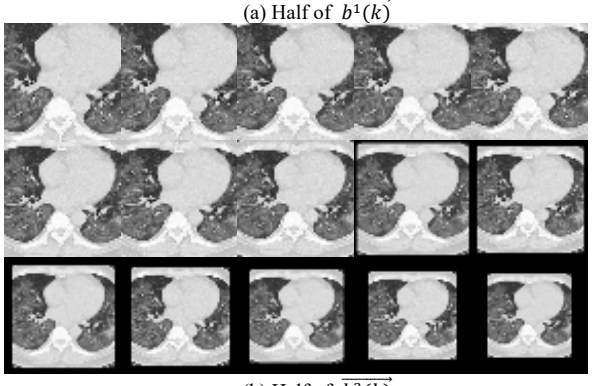
4.1 Result of Data Augmentation

Suppose $b(k)$ is Figure 2(a), Figure 6 shows the DA(1-7) results. Due to the page limit, their horizontal results DA(8-14) are not presented in this paper. Particularly, we only select 15 new generated images among 30 generate results per DA technique.

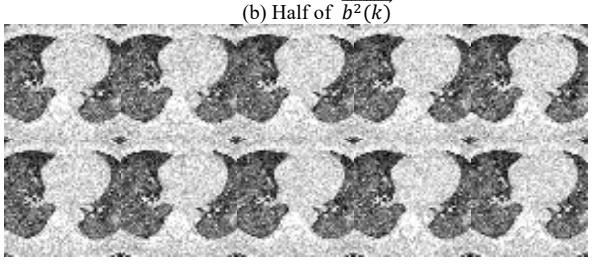
Figure 6(a) presents the 15 rotated new images. Figure 6(b-e) present 15 scaled, 15 noise-injected, 15 randomly translated, and 15 Gamma corrected images, respectively. Figure 6(f-g) present the 15 HST and 30 VST new images, respectively.



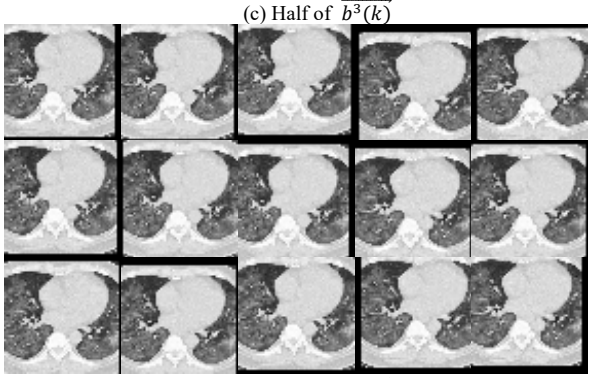
(a) Half of $\overrightarrow{b^1(k)}$



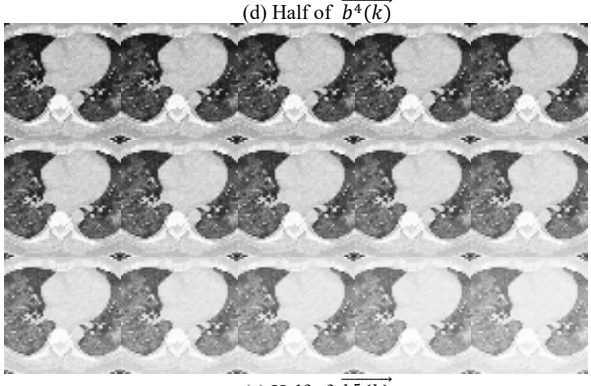
(b) Half of $\overrightarrow{b^2(k)}$



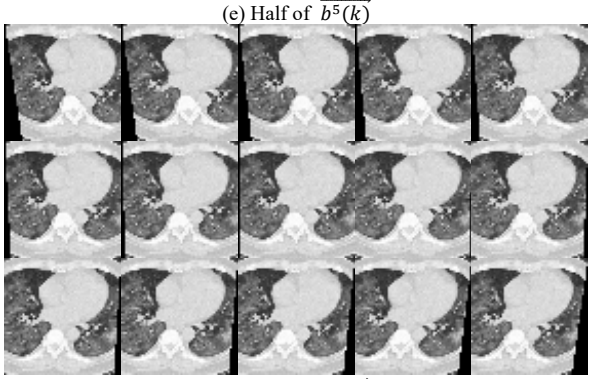
(c) Half of $\overrightarrow{b^3(k)}$



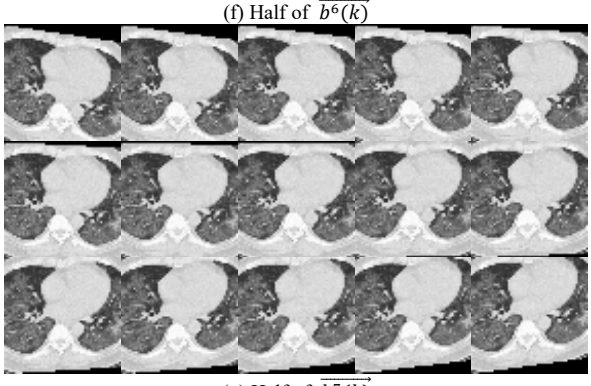
(d) Half of $\overrightarrow{b^4(k)}$



(e) Half of $\overrightarrow{b^5(k)}$



(f) Half of $\overrightarrow{b^6(k)}$



(g) Half of $\overrightarrow{b^7(k)}$

Figure 6 Half of DA(1-7) Results

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3025855, IEEE Sensors Journal

4.2 SP compared with other three pooling methods

The results of SP against other three pooling methods were presented in **Table 5**, which indicates that SP obtained the best sensitivity, accuracy, F1, MCC, and FMI. The definition of β can be found in Eqs (39.a)-(42).

For the specificity and precision indicators, the AP achieved the best performance. If we consider all the indicators, SP wins five out of seven indicators. Hence, SP gives the best performance compared to other three pooling methods.

Table 5 Ten runs of different pooling methods

L2P	β^1	β^2	β^3	β^4	β^5	β^6	β^7
1	90.63	94.69	94.46	92.66	92.50	85.38	92.52
2	91.56	92.81	92.72	92.19	92.14	84.38	92.14
3	92.50	94.06	93.97	93.28	93.23	86.57	93.23
4	93.13	93.75	93.71	93.44	93.42	86.88	93.42
5	92.19	93.44	93.35	92.81	92.77	85.63	92.77
6	91.56	93.44	93.31	92.50	92.43	85.01	92.43
7	93.13	94.38	94.30	93.75	93.71	87.51	93.71
8	93.75	91.56	91.74	92.66	92.74	85.33	92.74
9	93.13	95.31	95.21	94.22	94.15	88.46	94.16
10	93.75	93.75	93.75	93.75	93.75	87.50	93.75
M+SD	92.53	93.72	93.65	93.13	93.08	86.27	93.09
	±1.04	±1.04	±0.96	±0.66	±0.67	±1.31	±0.66
AP	β^1	β^2	β^3	β^4	β^5	β^6	β^7
1	91.25	94.38	94.19	92.81	92.70	85.67	92.71
2	91.88	94.06	93.93	92.97	92.89	85.96	92.90
3	92.50	92.19	92.21	92.34	92.36	84.69	92.36
4	92.81	94.69	94.59	93.75	93.69	87.52	93.70
5	92.81	95.00	94.89	93.91	93.84	87.83	93.84
6	91.25	92.50	92.41	91.88	91.82	83.76	91.83
7	92.50	92.50	92.50	92.50	92.50	85.00	92.50
8	93.44	95.31	95.22	94.38	94.32	88.77	94.33
9	92.81	94.38	94.29	93.59	93.54	87.20	93.55
10	95.63	94.38	94.44	95.00	95.03	90.01	95.03
M+SD	92.69	93.94	93.87	93.31	93.27	86.64	93.27
	±1.25	±1.12	±1.09	±0.98	±0.99	±1.96	±0.99
MP	β^1	β^2	β^3	β^4	β^5	β^6	β^7
1	94.69	95.31	95.28	95.00	94.98	90.00	94.98
2	92.19	92.81	92.77	92.50	92.48	85.00	92.48
3	94.69	94.69	94.69	94.69	94.69	89.38	94.69
4	93.75	92.81	92.88	93.28	93.31	86.57	93.31
5	92.50	94.38	94.27	93.44	93.38	86.89	93.38
6	95.31	91.56	91.87	93.44	93.56	86.94	93.57
7	94.38	93.44	93.50	93.91	93.93	87.82	93.94
8	95.00	94.69	94.70	94.84	94.85	89.69	94.85
9	94.06	93.13	93.19	93.59	93.62	87.19	93.62
10	94.38	94.38	94.38	94.38	94.38	88.75	94.38
M+SD	94.09	93.72	93.75	93.91	93.92	87.82	93.92
	±1.03	±1.15	±1.08	±0.80	±0.80	±1.60	±0.80
SP (Ours)	β^1	β^2	β^3	β^4	β^5	β^6	β^7
1	95.00	90.63	91.02	92.81	92.97	85.71	92.99
2	93.13	92.50	92.55	92.81	92.83	85.63	92.84
3	94.69	93.13	93.23	93.91	93.95	87.82	93.96
4	94.69	95.31	95.28	95.00	94.98	90.00	94.98
5	95.31	92.81	92.99	94.06	94.14	88.15	94.14
6	94.06	95.31	95.25	94.69	94.65	89.38	94.66
7	93.75	95.00	94.94	94.38	94.34	88.76	94.34
8	94.69	92.19	92.38	93.44	93.52	86.90	93.53
9	93.75	94.69	94.64	94.22	94.19	88.44	94.19
10	95.31	94.69	94.72	95.00	95.02	90.00	95.02
M+SD	94.44	93.63	93.70	94.03	94.06	88.08	94.06
	±0.73	±1.60	±1.47	±0.80	±0.76	±1.59	±0.76

4.3 Effect of DA

We compared using our 14-way DA "DA14" against not using DA (symbolized as DA0), to explore the effects of our DA14 strategies. The cognate comparison performance is presented in Table 6.

We can observe training with DA14 could significantly provide better performance than DA0 in terms of all seven indicators. Furthermore, the SD of results of DA14 are slightly smaller than that of DA0.

Table 6 Comparison of DA0 and DA14

DA	β^1	β^2	β^3	β^4	β^5	β^6	β^7
DA0	92.06	91.59	91.65	91.83	91.85	83.67	91.85
	±0.85	±1.60	±1.48	±0.96	±0.92	±1.92	±0.92
DA14	94.44	93.63	93.70	94.03	94.06	88.08	94.06
(Ours)	±0.73	±1.60	±1.47	±0.80	±0.76	±1.59	±0.76

4.4 Comparison to State-of-the-art methods

Our 7L-CNN-CD method was compared with five state-of-the-art approaches: RBFNN [6], K-ELM [7], ELM-BA [8], GoogLeNet [11], and ResNet18 [12].

All performances were compared on test set and presented in Table 7. Omitting the SD information, the comparison plot is presented in Figure 7, with measurement indicators chosen from β^1 to β^7.

Table 7 Comparison to state-of-the-art approaches

Approach	β^1	β^2	β^3	β^4	β^5	β^6	β^7
RBFNN[6]	67.08	74.48	72.52	70.78	69.64	41.74	69.64
K-ELM[7]	57.29	61.46	59.83	59.38	58.46	18.81	58.46
ELM-BA	57.08	72.40	67.48	64.74	61.75	29.90	61.76
[8]	±3.86	±3.03	±1.65	±1.26	±2.24	±2.45	±2.24
GoogLeNet	76.88	83.96	82.84	80.42	79.65	61.10	79.65
[11]	±3.92	±2.29	±1.58	±1.40	±1.92	±2.62	±1.91
ResNet18	78.96	89.48	88.30	84.22	83.31	68.89	83.32
[12]	±2.90	±1.64	±1.50	±1.23	±1.53	±2.33	±1.53
7L-CNN-CD	94.44	93.63	93.70	94.03	94.06	88.08	94.06
(Ours)	±0.73	±1.60	±1.47	±0.80	±0.76	±1.59	±0.76

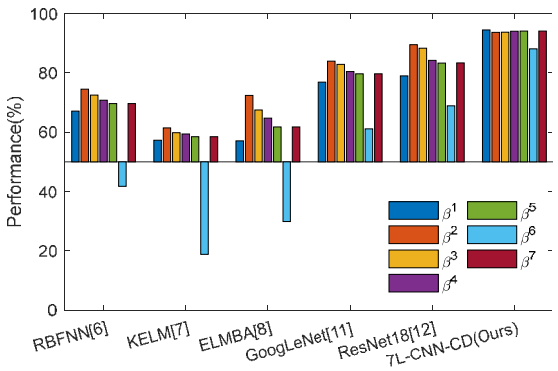


Figure 7 Bar plot of performances of six different methods

5 Conclusion

In this COVID-19 diagnosis study, a novel 7L-CNN-CD was proposed, using a seven-layer standard convolutional neural network as background, and integrating data augmentation and stochastic pooling methods.

Experimental results showcased our 7L-CNN-CD algorithm obtained excellent test performances: $\beta^1 = 94.44 \pm 0.73$,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2020.3025855, IEEE Sensors Journal

$\beta^2 = 93.63 \pm 1.60$, $\beta^3 = 93.70 \pm 1.47$, $\beta^4 = 94.03 \pm 0.80$, $\beta^5 = 94.06 \pm 0.76$, $\beta^6 = 88.08 \pm 1.59$, $\beta^7 = 94.06 \pm 0.76$. The results are better than five state-of-the-art algorithms in terms of COVID-19 diagnosis.

In our future studies, we shall attempt to (i) test more advanced data augmentation techniques; (ii) collect more COVID-19 data to test our algorithm; and (iii) move our algorithm to cloud computing platform to benefit radiologists.

Appendix

Table 8 Abbreviation list

Meanings	Abbreviations
MV	majority voting
SLS	Slice level selection
HC	Healthy control
CCT	Chest computed tomography
DS	downsampling
HS	histogram stretching
CR	compression ratio
DA	Data augmentation
(A)(M)(S)(L2)P	(Average) (Max) (Stochastic) (l_2 norm) pooling
MCC	Matthews correlation coefficient
FMI	Fowlkes–Mallows index

References

[1] D. Balsalobre-Lorente, O. M. Driha, F. V. Bekun, A. Sinha, and F. F. Adedoyin, "Consequences of COVID-19 on the social isolation of the Chinese economy: accounting for the role of reduction in carbon emissions," *Air Quality Atmosphere and Health*, p. 13.

[2] M. Chaudhary, P. R. Sodani, and S. Das, "Effect of COVID-19 on Economy in India: Some Reflections for Policy and Programme," *Journal of Health Management*, p. 12, Article ID: 0972063420935541.

[3] G. S. Campos, S. I. Sardi, M. B. Falcao, E. Belitardo, D. Rocha, C. A. Rolo*, et al.*, "Ion torrent-based nasopharyngeal swab metatranscriptomics in COVID-19," *Journal of Virological Methods,* vol. 282, p. 3, Article ID: 113888, Aug, 2020.

[4] A. Mahdavi, S. Haseli, A. Mahdavi, M. Bakhshayeshkaram, M. Foroumandi, S. M. Nekooghadam*, et al.*, "The Role of Repeat Chest CT Scan in the COVID-19 Pandemic," *Academic Radiology,* vol. 27, pp. 1049-1050, Jul, 2020.

[5] Y. Li and L. Xia, "Coronavirus Disease 2019 (COVID-19): Role of Chest CT in Diagnosis and Management," *AJR Am J Roentgenol,* vol. 214, pp. 1280-1286, Jun, 2020.

[6] Z. Lu, "A Pathological Brain Detection System Based on Radial Basis Function Neural Network," *Journal of Medical Imaging and Health Informatics,* vol. 6, pp. 1218-1222, 2016.

[7] J. Yang, "A pathological brain detection system based on kernel based ELM," *Multimedia Tools and Applications,* vol. 77, pp. 3715-3728, 2018.

[8] S. Lu, "A Pathological Brain Detection System based on Extreme Learning Machine Optimized by Bat Algorithm," *CNS & Neurological Disorders - Drug Targets,* vol. 16, pp. 23-29, 2017.

[9] 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.

[10] X. Jiang, "Chinese Sign Language Fingerspelling Recognition via Six-Layer Convolutional Neural Network with Leaky Rectified Linear Units for Therapy and Rehabilitation," *Journal of Medical Imaging and Health Informatics,* vol. 9, pp. 2031-2038, 2019.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov*, et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.

[12] X. Yu and S.-H. Wang, "Abnormality Diagnosis in Mammograms by Transfer Learning Based on ResNet18," *Fundamenta Informaticae,* vol. 168, pp. 219-230, 2019.

[13] Y. Zhang, Y. F. Qian, D. Wu, M. S. Hossain, A. Ghoneim, and M. Chen, "Emotion-Aware Multimedia Systems Security," *IEEE Transactions on Multimedia,* vol. 21, pp. 617-624, Mar, 2019.

[14] Y. Zhang, R. Gravina, H. M. Lu, M. Villari, and G. Fortino, "PEA: Parallel electrocardiogram-based authentication for smart healthcare systems," *Journal of Network and Computer Applications,* vol. 117, pp. 10-16, Sep, 2018.

[15] Y. Zhang, X. Ma, J. Zhang, M. S. Hossain, G. Muhammad, and S. U. Amin, "Edge Intelligence in the Cognitive Internet of Things: Improving Sensitivity and Interactivity," *IEEE Network,* vol. 33, pp. 58-64, May-Jun, 2019.

[16] M. Veluchamy and B. Subramani, "Fuzzy dissimilarity contextual intensity transformation with gamma correction for color image enhancement," *Multimedia Tools and Applications*, p. 17.

[17] J. M. Górriz, "Artificial intelligence within the interplay between natural and artificial computation: Advances in data science, trends and applications," *Neurocomputing,* vol. 410, pp. 237-270, 2020.

[18] M. Rezaei, H. Yang, and C. Meinel, "Deep Neural Network with l2-Norm Unit for Brain Lesions Detection," in *International Conference on Neural Information Processing (ICNIP)*, Cham, 2017, pp. 798-807.

[19] A. Ghosh, S. Singh, and D. Sheet, "Simultaneous Localization and Classification of Acute Lymphoblastic Leukemic Cells in Peripheral Blood Smears Using a Deep Convolutional Network with Average Pooling Layer," in *International Conference on Industrial and Information Systems*, Peradeniya, SRI LANKA, 2017, pp. 529-534.

[20] C. Huang, "Multiple Sclerosis Identification by 14-Layer Convolutional Neural Network With Batch Normalization, Dropout, and Stochastic Pooling," *Frontiers in Neuroscience,* vol. 12, Article ID: 818, 2018-November-08, 2018.

[21] X. W. Jiang, "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, pp. 15697-15715, Jun, 2020.

[22] S. Sun, B. Hu, Z. Yu, and X. N. Song, "A Stochastic Max Pooling Strategy for Convolutional Neural Network Trained by Noisy Samples," *International Journal of Computers Communications & Control,* vol. 15, p. 9, Article ID: Unsp 1007, Feb, 2020.

Prof. **Yu-Dong Zhang** (M'08, SM'15) received his BE in Information Sciences in 2004, and MPhil in Communication and Information Engineering in 2007, from Nanjing University of Aeronautics and Astronautics. He received the PhD degree in Signal and Information Processing from Southeast University in 2010. Now he serves as Professor with University of Leicester.

Dr Suresh Chandra Satapathy is a Ph.D. in Computer Science Engineering, currently working as Professor of School of Computer Engg and Dean- Research at KIIT, Bhubaneshwar, Odisha, India. He has developed two new optimization algorithms, Social Group Optimization (SGO) published in Springer Journal and SELO (Social Evolution and Learning Algorithm) published in Elsevier.

Professor **Liyao Zhu** received a bachelor's degree in clinical medicine from Xuzhou Medical University in 1989. He has presided over 5 municipal scientific research projects, published more than 50 papers, including 7 SCI articles, and co-edited a monograph. He has won 3 Municipal Scientific and Technological Progress Awards and 3 Municipal New Technology Introduction Awards.

Prof. **J.M. Górriz** received the B.Sc. degree in physics, the B.Sc. degree in electronic engineering from the University of Granada, Spain, in 2000 and 2001, respectively, Ph.D. degree from the University of Cádiz, Spain, in 2003, and the Ph.D. degree from the University of Granada in 2006. He is currently a Full Professor with University of Granada.

Dr. Shui-Hua Wang (M'11-SM'20) got her bachelor's degree in Information Sciences from Southeast University in 2008, master's degree in Electrical Engineering from City college of New York in 2012, and Ph.D. degree in Electrical Engineering from Nanjing University in 2017. Now, she is working as Research Associate in University of Leicester, UK.