Machine Vision and Applications (2021) 32:14
https://doi.org/10.1007/s00138-020-01128-8

SPECIAL ISSUE PAPER

A five-layer deep convolutional neural network with stochastic pooling for chest CT-based COVID-19 diagnosis

Yu-Dong Zhang[1,2] · Suresh Chandra Satapathy[3] · Shuaiqi Liu[4] · Guang-Run Li[5]

Received: 11 July 2020 / Revised: 4 September 2020 / Accepted: 15 September 2020

Abstract

Till August 17, 2020, COVID-19 has caused 21.59 million confirmed cases in more than 227 countries and territories, and 26 naval ships. Chest CT is an effective way to detect COVID-19. This study proposed a novel deep learning model that can diagnose COVID-19 on chest CT more accurately and swiftly. Based on traditional deep convolutional neural network (DCNN) model, we proposed three improvements: (i) We introduced stochastic pooling to replace average pooling and max pooling; (ii) We combined conv layer with batch normalization layer and obtained the conv block (CB); (iii) We combined dropout layer with fully connected layer and obtained the fully connected block (FCB). Our algorithm achieved a sensitivity of $93.28\% \pm 1.50\%$, a specificity of $94.00\% \pm 1.56\%$, and an accuracy of $93.64\% \pm 1.42\%$, in identifying COVID-19 from normal subjects. We proved using stochastic pooling yields better performance than average pooling and max pooling. We compared different structure configurations and proved our 3CB + 2FCB yields the best performance. The proposed model is effective in detecting COVID-19 based on chest CT images.

Keywords Deep convolutional neural network · Stochastic pooling · COVID-19 · Batch normalization · Dropout · Convolution block · Fully connected block

1 Introduction

The coronavirus pandemic is an ongoing global pandemic disease, which is also called COVID-19. World Health Organization (WHO) declared the COVID-19 as a public health crisis of global concern on 30/01/2020, and as a pandemic on 11/03/2020 [1]. Till August 17, 2020, COVID-19 has caused 21.59 million confirmed cases and 773.6 thousand death tolls.

Recommended preventive measures are composed of mouth covering when coughing, hand washing, social distancing, face masks in public, suspect isolation, etc. From the viewpoint of countries, lockdown, travel restriction, facility closure, workplace control, contact tracing, testing capacity increase are all effective preventive measures.

Reverse transcription polymerase chain reaction (RT-PCR) [2] and real-time RT-PCR [3] are one of the standard diagnosis methods from a nasopharyngeal swab. Chest computed tomography (CCT) is another effective diagnosis tool for COVID-19 diagnosis. Compared to polymerase chain reaction (PCR), CCT is quicker and more sensitive [4]. The main biomarkers differentiating COVID-19 from healthy people are the asymmetric peripheral ground-glass opacities (GGOs) without pleural effusions [5]. Manual interpreta-

Yu-Dong Zhang and Suresh Chandra Satapathy have contributed equally to this work.

✉ Shuaiqi Liu
shdkj-1918@163.com

✉ Guang-Run Li
3046322112@qq.com

Yu-Dong Zhang
yudongzhang@ieee.org

Suresh Chandra Satapathy
sureshsatapathy@ieee.org

1 School of Informatics, University of Leicester, Leicester LE1 7RH, UK

2 Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia

3 School of Computer Engg, KIIT Deemed To University, Bhubaneswar, India

4 College of Electronic and Information Engineering, Hebei University, Baoding 071002, Hebei, China

5 Department of Imaging, Jinhu People's Hospital, No. 160, Jinhu Shenhua Street, Huai'an, China

Published online: 03 November 2020

tion by radiologists is tedious and easy to be influenced by fatigue, emotion, and other factors. A smart diagnosis system via computer vision and artificial intelligence can benefit patients, radiologists, 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. Jiang [9] used a six-layer convolutional neural network to recognize sign language fingerspelling. Their method is abbreviated as 6L-CNN-F, here F means fingerspelling. Szegedy et al. [10] presented the GoogLeNet. Yu and Wang [11] suggested the use of ResNet18 for mammogram abnormality detection. Two references provide systematic reviews of machine learning techniques in detecting COVID-19 [12, 13]. Besides, there are some successful applications in other industrial and academic fields using traditional AIs [14–18].

This study used deep convolutional neural network (DCNN) as the backbone. To make our algorithm effective in detecting COVID-19, we proposed three improvements, (i) We introduced stochastic pooling (SP) to replace traditional average pooling and maximum pooling methods; (ii) We created conv block (CB) by combining conv layer and batch normalization, and (iii) we created fully connected block (FCB) by combining dropout layer and fully connected layer.

Those three improvements help enrich the performance of the basic DCNN, and we name our proposed algorithm as "5-layer DCNN with stochastic pooling for COVID-19 (5L-DCNN-SP-C) algorithm." Sections 2, 3, 4, and 5 present the dataset, methodology, results, and conclusions, respectively.

2 Dataset

We enrolled 142 COVID-19 subjects and 142 healthy controls (HCs) from local hospitals. CCT was performed on all subjects, and three-dimensional volumetric images were obtained. Slice level selection (SLS) method was used: For COVID-19 pneumonia patients, the slice showing the largest size and number of lesions was selected. For healthy controls, any level of the image can be selected. Use this slice level selection method, we extract 320 images (resolution: 1024×1024) from both COVID-19 patients and HC subjects, respectively. The demographics of our image set are offered in Table 1. Table 2 shows the abbreviation list for easy reading.

Table 1 Demographics of COVID-19 and HC

	No. Subjects	No. Images	Age Range
COVID-19	142	320	22–91
HC	142	320	21–76

Table 2 Abbreviation list

Meanings	Abbreviations
CCT	Chest computed tomography
BCR	Byte compression ratio
SLS	Slice level selection
NLAF	Nonlinear activation function
AM	Activation map
(A)(M)(S)P	(average) (max) (stochastic) pooling
NLDS	nonlinear downsampling
DW	Down-weight
DO(L)(N)	Dropout (layer) (neuron)
CRLW	Compression ratio of learnable weights
PL	Pooling layer
SC	Structure configuration
CB	Convolution block
FCB	Fully connected block

3 Methodology

3.1 Preprocessing

Let us set the original CCT image set to be S_1, which is composed of n CCT images as

$$S_1 = \{s_1(1), s_1(2), \ldots, s_1(i), \ldots s_1(n)\}. \tag{1}$$

First, we compress the three-channel color image to gray image, and get the grayscale image set S_2 as

$$\begin{aligned} S_2 &= G(S_1|\text{RGB} \rightarrow \text{Grayscale}) \\ &= \{s_2(1), s_2(2), \ldots, s_2(i), \ldots, s_2(n)\}. \end{aligned} \tag{2}$$

Second, the histogram stretching (HS) method was firstly employed to increase the image's contrast. For i-th image $s_2(i)$, the new histogram stretched image $s_3(i)$ was obtained as

$$s_3(i|\alpha, \beta) = \frac{s_2(i|\alpha, \beta) - \varepsilon_2^{\min}(i)}{\varepsilon_2^{\max}(i) - \varepsilon_2^{\min}(i)}, \tag{3}$$

where $1 \le \alpha \le 1024$, $1 \le \beta \le 1024$. Here, (α, β) means coordinates of pixel of the image $s_2(i)$, and $\varepsilon_2^{\min}(i)$ means

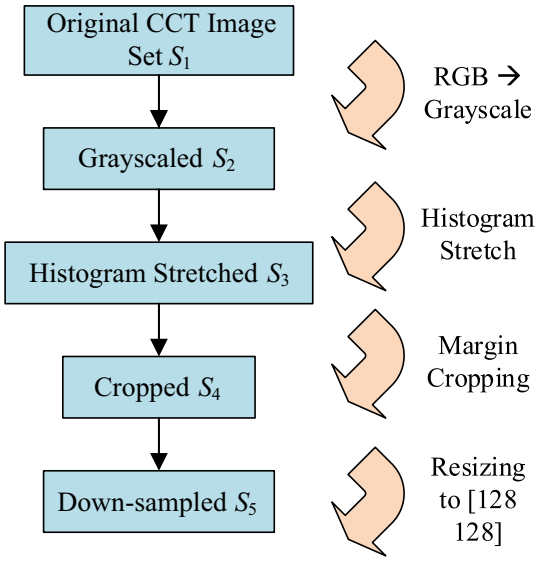


Fig. 1 Diagram of preprocessing (color figure online)

the minimum value of CCT image $s_2(i)$. $\varepsilon_2^{\max}(i)$ means the maximum value of image $s_2(i)$.

$$\varepsilon_2^{\min}(i) = \min_{(\alpha,\beta)}[s_2(i|\alpha,\beta)] \tag{4a}$$

$$\varepsilon_2^{\max}(i) = \max_{(\alpha,\beta)}[s_2(i|\alpha,\beta)]. \tag{4b}$$

In all, we get the histogram stretched dataset S_3 as

$$\begin{aligned} S_3 &= \mathrm{HS}(S_2) \\ &= \{s_3(1), s_3(2), \ldots, s_3(i), \ldots s_3(n)\}. \end{aligned} \tag{5}$$

Third, we crop the images to remove the texts at the margin area, and the checkup bed at the bottom area. Thus, we get the cropped dataset S_4 as

$$\begin{aligned} S_4 &= C\big(S_3, [\mathrm{top}, \mathrm{bottom}, \mathrm{left}, \mathrm{right}]\big) \\ &= \{s_4(1), s_4(2), \ldots, s_4(i), \ldots s_4(n)\}, \end{aligned} \tag{6}$$

where C represents crop operation, and the parameter vector [top, bottom, left, right] means to the range to be removed from top, bottom, left, and right directions. In our study, we set top = bottom = left = right = 150.

Fourth, we downsampled the image $s_4(i)$ to size of $[\varpi, \varpi]$, and we now get the resized image set S_5 as

$$\begin{aligned} S_5 &= \updownarrow(S_4, [\varpi, \varpi]) \\ &= \{s_5(1), s_5(2), \ldots, s_5(i), \ldots s_5(n)\} \end{aligned} \tag{7}$$

where ↕means downsampling operation. $\varpi = 128$ in this study. Figure 1 shows the above four preprocessing steps.

Table 3 compares the size and storage per image at every preprocessing step. We can see here after the five-step preprocessing procedure, each image will only cost about 0.52% of its original storage. The byte compression ratio (BCR) was calculated as: BCR = byte(s_5) ÷ byte(s_1) = 65, 536 ÷ 12, 582, 912 = 0.52%.

Figure 2 shows two samples of our collected and preprocessed dataset S_5, from which we can clearly observe the clinical biomarkers of COVID-19. Cui et al. [19] reported the preliminary CT findings of COVID-19 in their publication. Tuncer et al. [20] used chest CT images, and then developed a local binary pattern and iterative ReliefF algorithm. There are more open publications that show it is feasible to develop effective AI systems based on CCT images.

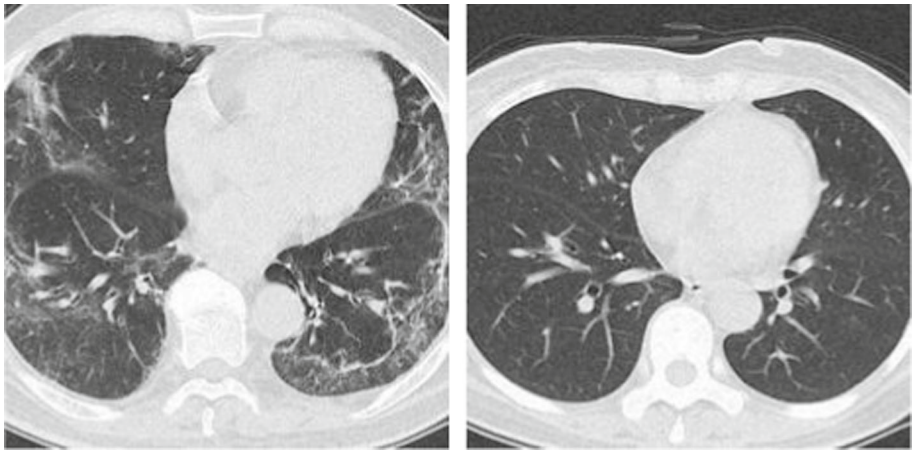
3.2 Basics of DCNN

Deep convolutional neural network (DCNN) is a king of new artificial neural network. Its main feature is to use multiple layers to build a deep neural network. Generally, DCNN is composed of conv layers (CLs), pooling layers (PLs), and fully connected layers (FCLs) [21–25]. Figure 3 presents a simplistic instance consisting of 2 CLs, 2 PLs, and 2 FCLs. On the right part of Fig. 3, The blue rectangle means FCL block, and red rectangle means the softmax function. DCNNs could reach better performances than old-dated AI methods, because they learn the feature from the data during the training procedure. There is no need to consume much time in feature engineering.

The essential operation in DCNN is convolution. The CL performed 2D convolution along the width and height directions. Note that the weights in CNN are initialized with random, and then learnt from data itself by network training. Figure 4 illustrates the pipeline of input feature maps passing across a CL. Assume there is an input matrix, J kernels (K_1,

Table 3 Image size and storage per image at each preprocessing step

Preprocessing step	Image Size (per image)	Byte(s) (per image)
Original s_1	$1024 \times 1024 \times 3 = 3, 145, 728$	12,582,912
Grayscaled s_2	$1024 \times 1024 \times 1 = 1, 048, 576$	4,194,304
Histogram stretched s_3	$1024 \times 1024 \times 1 = 1, 048, 576$	4,194,304
Cropped s_4	$724 \times 724 \times 1 = 524, 176$	2,096,704
Downsampled s_5	$128 \times 128 \times 1 = 16, 384$	65,536



(a) COVID-19 **(b)** HC

Fig. 2 Two samples of our preprocessed dataset S_5

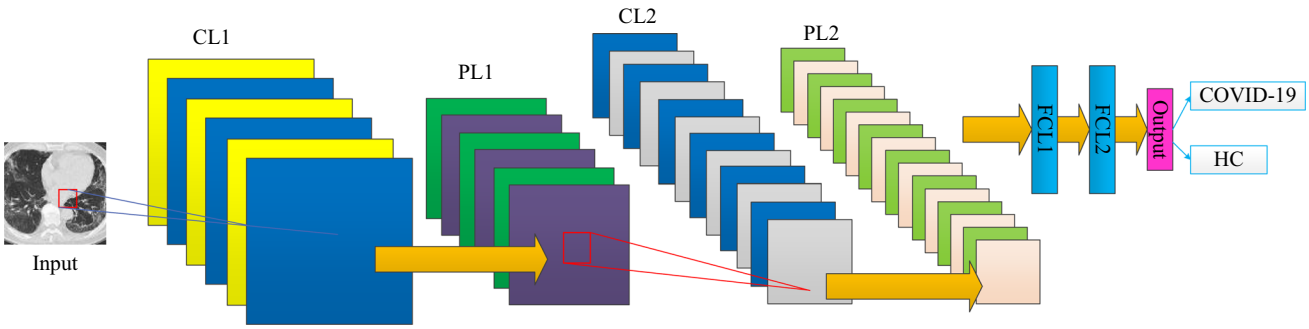


Fig. 3 Pipeline of a toy example of DCNN with 2 CLs, 2PLs, and 2 FCLs

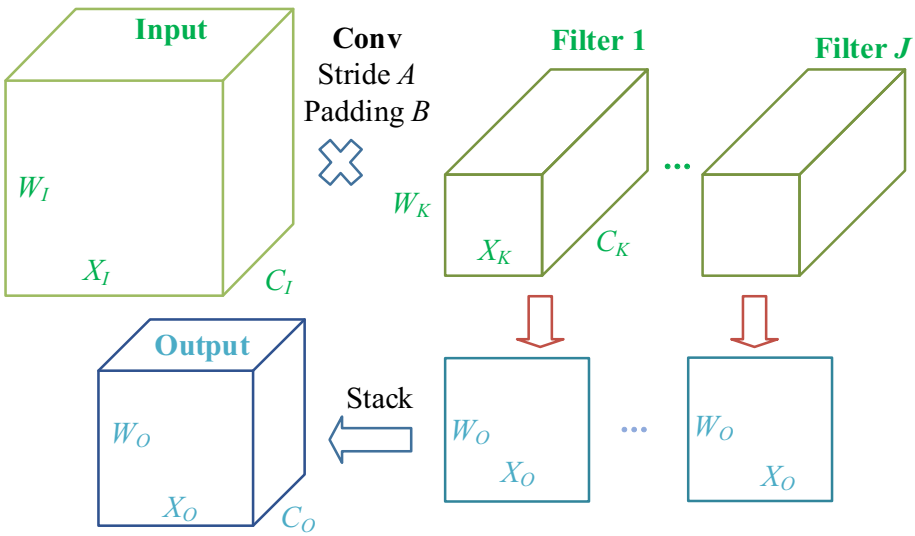


Fig. 4 Pipeline of conv layer

$K_2, \ldots, K_j, \ldots, K_J$), and an output O, with theirs sizes $\mathbb{S}$ defined as

$$\mathbb{S}(x) = \begin{cases} W_I \times X_I \times C_I & x = I \\ W_K \times X_K \times C_K & x = K_j (j = 1, \ldots, J), \\ W_O \times X_O \times C_O & x = O \end{cases} \tag{8}$$

where (W, X, C) represent the size of height, width, and channels of the matrix, respectively. Subscript I, K, and O represent input, kernel, and output, respectively. J denotes total number of filters. Note that

$$C_I = C_K \tag{9a}$$

$$C_O = J \tag{9b}$$

which means the channel of input C_I should equal the channel of kernel C_K, and the channel of output C_O should equal the number of filters J.

Assume those filters move with padding of B and stride of A, we can get their relationship by simple math as:

$$W_O = 1 + \frac{(2 \times B + W_I - W_K)}{A} \tag{10a}$$

$$X_O = 1 + \frac{(2 \times B + X_I - X_K)}{A}, \tag{10b}$$

where $\lfloor . \rfloor$ represents the floor function. Afterward, CL's outputs are hurled into a nonlinear activation function (NLAF) σ, that usually selects the rectified linear unit (ReLU) function.

$$\begin{aligned} \sigma_{\mathrm{ReLU}}(x) &= \mathrm{ReLU}(x) \\ &= \max(0, x). \end{aligned} \tag{11}$$

ReLU is preferred to traditional NLAFs such as hyperbolic tangent (HT) and sigmoid (SM) function

$$\begin{aligned} \sigma_{\mathrm{HT}}(x) &= \tanh(x) \\ &= \frac{\left(e^x - e^{-x}\right)}{\left(e^x + e^{-x}\right)} \end{aligned} \tag{12}$$

$$\sigma_{\mathrm{SM}}(x) = \left(1 + e^{-x}\right)^{-1}. \tag{13}$$

3.3 Improvement 1: Use SP to replace MP and AP

The activation maps (AMs) after each block within DCNN are usually too large, i.e., the size of their width, length, and channels are too large to handle, which will cause (i) overfitting of the training and (ii) large computational costs.

Pooling layer (PL) is a form of nonlinear downsampling (NLDS) method to solve above issue. Further, PL can provide invariance-to-translation property to the AMs. For a 2×2 region, suppose the pixels within the region $\overline{\varphi}$ are

$$\overline{\varphi} = \begin{bmatrix} \varphi_{1,1} & \varphi_{1,2} \\ \varphi_{2,1} & \varphi_{2,2} \end{bmatrix}. \tag{14}$$

The average pooling (AP) calculates the mean value in the region $\overline{\varphi}$. Assume the output value after NLDS is z, we can have

$$\begin{aligned} z_{\overline{\varphi}}^{AP} &= \mathrm{average}(\overline{\varphi}) \\ &= \frac{\varphi_{1,1} + \varphi_{1,2} + \varphi_{2,1} + \varphi_{2,2}}{|\overline{\varphi}|}, \end{aligned} \tag{15}$$

where $|\overline{\varphi}|$ means the number of elements of region $\overline{\varphi}$. Here, $|\overline{\varphi}| = 4$ if we use a 2×2 NLDS pooling. Using Fig. 5 as an example, and assuming the region $\dot{\varphi}$ at 2^{nd} row 1^{st} column of the input AM, I is chosen, i.e., $\dot{\varphi} = I(row = 2, col = 1)$; thus, we have $z_{\dot{\varphi}}^{AP} = \mathrm{average}(\dot{\varphi}) = (4 + 4 + 3 + 9) \div 4 = 20 \div 4 = 5$.

The max pooling (MP) operates on the region $\overline{\varphi}$ and selects the max value. Note that both AP and MP work on every slice separately.

$$\begin{aligned} z_{\overline{\varphi}}^{MP} &= \max(\overline{\varphi}) \\ &= \max_{i,j=1}^{2} \varphi_{i,j}. \end{aligned} \tag{16}$$

In Fig. 5, $z_{\dot{\varphi}}^{MP} = \max(\dot{\varphi}) = \max(4 + 4 + 3 + 9) = 9$.

In practice, scholars observed that the AP did not work well, because all pixels in the region $\overline{\varphi}$ are within the arguments of the NLDS function; hence, it could down-weight (DW) intense activation owing to numerous near-zero pixels. For example, in our region $\dot{\varphi}$, the strongest value $9 \overset{\mathrm{DM}}{\rightarrow} 5$. On the other hand, MP deciphers above DW problem; however, it simply overfits the training set and causes the lack-of-generalization (LoG) problem.

The stochastic pooling (SP) was introduced to conquer the DW, overfitting, and LoG problems caused by MP and AP. Instead of computing the average or the max, the output of the SP z^{SP} is calculated via sampling from a multinomial distribution generated from the activations of each region $\overline{\varphi}$. Three steps of SP are depicted below:

(1) Estimate the probability $\theta_{i,j} \in \Theta$ of each entry $\{\varphi_{i,j},\ i,\ j = 1,\ 2\}$ within the region $\overline{\varphi}$.

$$\theta_{i,j} = \frac{\varphi_{i,j}}{\mathrm{sum}(\overline{\varphi})}, \quad i, j = 1, 2 \tag{17a}$$

$$\sum_{i,j=1}^{2} \theta_{i,j} = 1 \tag{17b}$$

in which, (i, j) is the element index of region $\overline{\varphi}$. In matrix format, equation (17a) can be rewritten as

$$\Theta = \overline{\varphi} / \sum(\overline{\varphi}). \tag{18}$$

(2) Select a location β within $\overline{\varphi}$ in accordance with the probability $\{\theta_{i,j}\}$.

$$\beta \sim \mathrm{Prob}\left(\theta_{1,1}, \theta_{1,2}, \theta_{2,1}, \theta_{2,2}\right). \tag{19}$$

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

$$z_{\overline{\varphi}}^{\mathrm{SP}} = \varphi_{\beta} \tag{20}$$

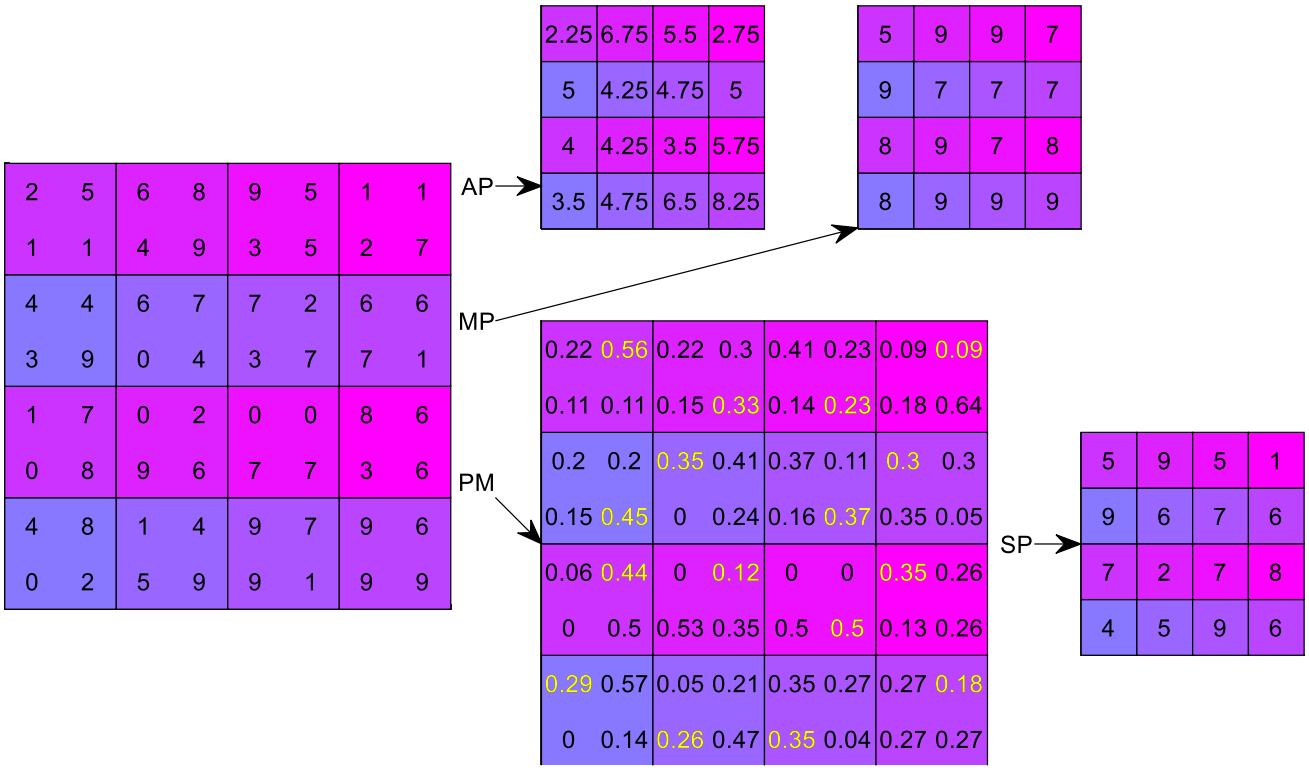


Fig. 5 Toy examples of different pooling technologies

Use the region $\dot{\varphi}$ in Fig. 5 as example, SP first calculates the probability map (PM),

$$\Theta(\dot{\varphi}) = \begin{bmatrix} 4 & 4 \\ 3 & 9 \end{bmatrix} / \sum\left(\begin{bmatrix} 4 & 4 \\ 3 & 9 \end{bmatrix}\right) = \begin{bmatrix} 0.2 & 0.2 \\ 0.15 & 0.45 \end{bmatrix} \tag{21a}$$

$$\beta(\dot{\varphi}) = (2, 2). \tag{21b}$$

Using the probability map, we randomly select the position $\beta = (2, 2)$ associates with probability of $\theta_{2,2}(\dot{\varphi}) = 0.45$. Thus, the SP output of $\dot{\varphi}$ is $z_{\dot{\varphi}}^{\mathrm{SP}} = \dot{\varphi}_{\beta} = \dot{\varphi}_{(2,2)} = 9$. In all, SP uses non-maximal activations from the region $\overline{\varphi}$, instead of outputting the greatest value.

3.4 Improvement 2: batch normalization transform

The motivation of batch normalization transform (BNT) is the so-called internal covariant shift (ICS), which means the effect of randomness of the distribution of inputs to internal DCNN layers during training. The phenomenon of ICS will worsen the DCNN's performance.

This study introduced BNT to normalize those internal layer's inputs $A = \{a_i\}$ over every mini-batch (suppose its size is m), in order to guarantee the batch normalized output $B = \{b_i\}$ have a uniform distribution. Mathematically, BNT is to learn a function from

$$\underbrace{\{a_i, i = 1, 2, \ldots, m\}}_{A} \mapsto \underbrace{\{b_i, i = 1, 2, \ldots, m\}}_{B}. \tag{22}$$

The empirical mean μ and empirical variance σ^2 over training set A can be calculated as

$$\mu_A = \frac{1}{m}\left(\sum_{i=1}^{m} a_i\right) \tag{23}$$

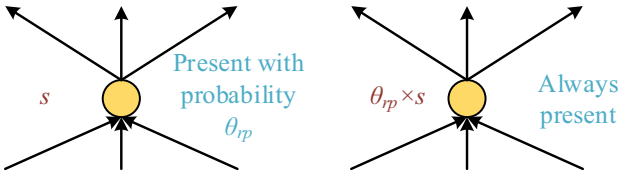
$$\sigma_A^2 = \frac{1}{m}\sum_{i=1}^{m}(a_i - \mu_A)^2. \tag{24}$$

The input $a_i \in A$ was first normalized to $\breve{a}_i$

$$\breve{a}_i = \frac{(a_i - \mu_A)}{\sqrt{(\sigma_A^2 + \Delta)}} \tag{25}$$

where Δ in denominator in Eq. (25) is to enhance the numerical stability. The value of Δ is a small constant. $\Delta = 10^{-5}$ in this study. Now has zero-mean and unit-variance characteristics. In order to have a more expressive deep neural network [26], a transformation is usually carried out as

$$b_i = \mathfrak{C} \times \breve{a}_i + \mathfrak{D}, i = 1,2,\cdots,m \tag{26}$$



(a) Training **(b)** Inference

Fig. 6 DONs at training and inference stages (s = weights, θ_{rp}=retention probability)

where the parameters $\mathfrak{C}$ and $\mathfrak{D}$ are two learnable parameters during training. The transformed output $b_i \in B$ is then passed to the next layer and the normalized remains internal to current layer.

In the inference stage, we do not have mini-batch anymore. So instead of calculating empirical mean and empirical variance, we will calculate population mean $\underline{\mu}$ and population variance $\underline{\sigma^2}$, and we have the output $\underline{b_i}$ at inference stage as

$$\underline{b_i} = \mathfrak{C} \times \left(\frac{a_i - \underline{\mu}}{sqrt\left(\underline{\sigma^2} + \Delta\right)} \right) + \mathfrak{D}. \tag{27}$$

We proposed to use convolution block (CB) to be one of the building blocks of our DCNN. The CB consists of one conv layer and one batch normalization layer.

3.5 Improvement 3: fully connected block

In traditional DCNN, the fully connected layer (FCL) serves the role of classifier. We plan to replace FCL with fully connected block (FCB), which will include one dropout layer (DOL) and one FCL layer. Srivastava et al. [27] proposed the concept of dropout neurons (DON) and DOL by randomly drop neurons and set to zero their neighboring weights s from the DCNN during training.

The neuron's incoming and outgoing connections are freezing, after it is dropped out. Figure 6 illustrates the illustration of neurons in DOL. The selections of dropout are random with a retention probability (θ_{rp}).

$$\underset{\text{training}}{\tilde{s}} = \begin{cases} s & \text{with } \theta_{rp} \\ 0 & \text{otherwise} \end{cases}. \tag{28}$$

where $\theta_{rp} = 0.5$, and $\tilde{s}$ means the weights of dropped out neurons.

During inference, we run the entire DCNN without dropout, but the weights of FCLs of FCBs are downscaled (viz., multiplied) by θ_{rp}.

$$\underset{\text{inference}}{\tilde{s}} = \theta_{rp} \times s. \tag{29}$$

Figure 7 shows a toy DCNN example with four FCL layers. Suppose we have $N(k)$ neurons at k-th layer, and assume $N(1) = 12$, $N(2) = 10$, $N(3) = 8$, $N(4) = 4$. Thus, we have in total $\sum_{k=1}^{4} N(k) = 34$ nodes. Suppose we do not consider incoming and outgoing weights, and do not consider the number of biases, the size of learnable weights $S^b(i, j)$ as number of weights between layer i and layer j before dropout, roughly calculating, can be written as $S^b(1, 2) = 12 \times 10 = 120$, $S^b(2, 3) = 10 \times 8 = 80$, $S^b(3, 4) = 8 \times 4 = 32$. In total, we have the total number of learnable weights before dropout as $S^b = \sum_{k=1}^{3} S^b(k, k+1) = 232$. Using $\theta_{rp} = 0.5$, the size of learnable weights after dropout between layer i and layer j is symbolized as $S^a(i, j)$, and we can calculate the total number of learnable weights as $S^a = \sum_{k=1}^{3} S^a(k, k+1) = S^a(1, 2) + S^a(2, 3) + S^a(3, 4) = 30 + 20 + 8 = 58$.

The compression ratio of learnable weights (CRLW), roughly, can be calculated by $58/232 = 0.25$, which is the squared value of retention probability θ_{rp}.

$$\text{CRLW} = \frac{S^a}{S^b} = \theta_{rp}^2, \tag{30}$$

where S^a and S^b means the number of learnable weights after and before dropout, respectively.

3.6 Proposed DCNN and its Implementation

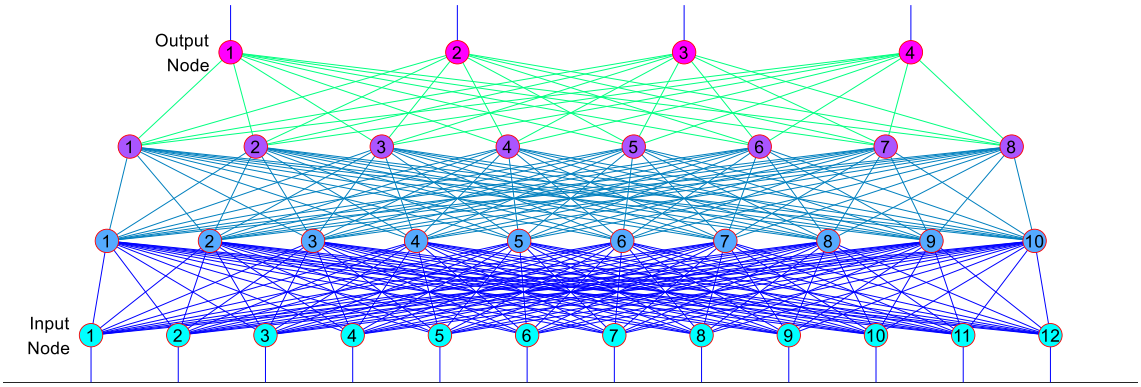
We create a new five-layer DCNN with stochastic pooling for COVID-19 detection (5L-DCNN-SP-C) with three CBs and two FCBs. The structure of proposed 5L-DCNN-SP-C is shown in Fig. 8, where SP is added after each activation map. The reason why set three CBs and two FCBs are by manual trial-and-error method. In the experiment, we will compare this setting (3 CBs + 2 FCBs) against other setting.

The hyperparameters of each layer/block of proposed 5L-DCNN-SP-C are listed in Table 4, where ($\alpha\beta \times \beta/\gamma$) means α filters with size of $\beta \times \beta$, followed by pooling layer with pooling size of γ. Meanwhile, W and B represent the size of weight matrix and bias vector, respectively. The last column in Table 4 shows the activation map (AM).

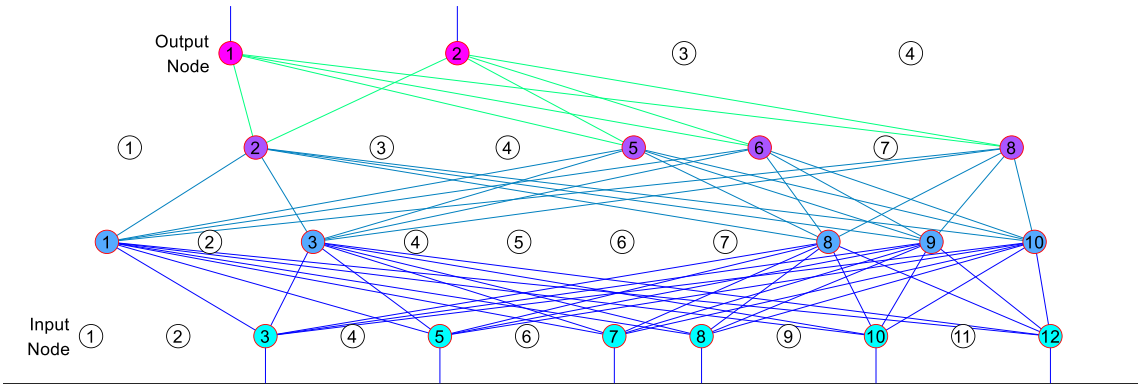
Ten runs of tenfold cross-validation were employed. Suppose confusion matrix $\mathbb{C}$ is defined as

$$\mathbb{C}(k, r) = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}, \tag{31}$$

where (c_{11}, c_{12}, c_{21}, c_{22}) represent TP, FN, FP, and TN, respectively. k is the index of trial (in each trial, onefold was used as test, and all the other folds were used as training), and r is the index of run.



(a) Before DO



(b) After DO

Fig. 7 A toy example of a DCNN with four FCLs

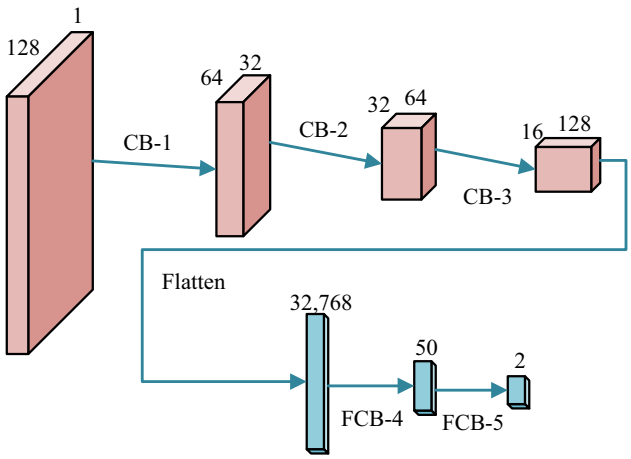


Fig. 8 Structure of proposed 5L-DCNN-SP-C

Table 4 Details of each layer in proposed 5L-DCNN-SP-C

Layer/Block	Hyperparameter	AM
Input	n/a	$128 \times 128 \times 1$
CB-1-SP	32 3×3 /2	$64 \times 64 \times 32$
CB-2-SP	64 3×3 /2	$32 \times 32 \times 64$
CB-3-SP	128 3×3 /2	$16 \times 16 \times 128$
Flatten		$1 \times 32{,}768$
FCB-4	W($50 \times 32{,}768$); B(50×1); $\theta_{rp} = 0.5$	1×50
FCB-5	W(2×50); B(2×1); $\theta_{rp} = 0.5$	1×2

n/a not available,*AM* activation map

Note that $\mathbb{C}$ will be calculated based on each test fold, and summarized across all 10 trials. Then, we get the

$$\mathbb{C}(r) = \sum_{k=1}^{10} \mathbb{C}(k, r). \tag{32}$$

Now we can calculate six indicators $\vec{\eta}(r)$ based on the confusion matrix over r-th run $\mathbb{C}(r)$.

$$\mathbb{C}(r) \mapsto \underbrace{[\eta_1(r), \eta_2(r), \ldots, \eta_6(r)]}_{\vec{\eta}(r)}, \tag{33}$$

Table 5 Ten runs of AP, MP, and SP

AP	η_1	η_2	η_3	η_4	η_5	η_6
1	93.13	90.00	90.30	91.56	91.69	83.17
2	90.94	90.00	90.09	90.47	90.51	80.94
3	92.50	90.94	91.08	91.72	91.78	83.45
4	90.94	92.19	92.09	91.56	91.51	83.13
5	91.56	90.63	90.71	91.09	91.14	82.19
6	91.88	92.50	92.45	92.19	92.16	84.38
7	90.94	90.63	90.65	90.78	90.80	81.56
8	92.19	89.38	89.67	90.78	90.91	81.59
9	88.75	90.31	90.16	89.53	89.45	79.07
10	89.38	88.13	88.27	88.75	88.82	77.51
Mean ± SD	91.22 ± 1.35	90.47 ± 1.27	90.55 ± 1.19	90.84 ± 1.05	90.88 ± 1.06	81.70 ± 2.10
MP	η_1	η_2	η_3	η_4	η_5	η_6
1	90.63	92.50	92.36	91.56	91.48	83.14
2	92.19	92.19	92.19	92.19	92.19	84.38
3	93.44	93.13	93.15	93.28	93.29	86.56
4	93.75	94.38	94.34	94.06	94.04	88.13
5	93.44	93.13	93.15	93.28	93.29	86.56
6	92.81	92.19	92.24	92.50	92.52	85.00
7	91.88	91.56	91.59	91.72	91.73	83.44
8	91.56	91.88	91.85	91.72	91.71	83.44
9	91.25	94.06	93.89	92.66	92.55	85.35
10	92.81	92.50	92.52	92.66	92.67	85.31
Mean ± SD	92.38 ± 1.04	92.75 ± 0.92	92.73 ± 0.89	92.56 ± 0.81	92.55 ± 0.82	85.13 ± 1.61
SP	η_1	η_2	η_3	η_4	η_5	η_6
1	91.25	91.56	91.54	91.41	91.39	82.81
2	95.31	94.69	94.72	95.00	95.02	90.00
3	93.75	95.94	95.85	94.84	94.79	89.71
4	91.25	94.06	93.89	92.66	92.55	85.35
5	95.00	96.25	96.20	95.63	95.60	91.26
6	92.50	92.81	92.79	92.66	92.64	85.31
7	92.19	91.88	91.90	92.03	92.04	84.06
8	95.00	94.38	94.41	94.69	94.70	89.38
9	93.13	93.75	93.71	93.44	93.42	86.88
10	93.44	94.69	94.62	94.06	94.03	88.13
Mean ± SD	93.28 ± 1.50	94.00 ± 1.56	93.96 ± 1.54	93.64 ± 1.42	93.62 ± 1.42	87.29 ± 2.83

where η_1 is sensitivity, η_2 is specificity, η_3 is precision, and η_4 is accuracy. Ignoring variable r, we have:

$$\eta_1 = \frac{c_{11}}{c_{11} + c_{12}} \tag{34a}$$

$$\eta_2 = \frac{c_{22}}{c_{22} + c_{21}} \tag{34b}$$

$$\eta_3 = \frac{c_{11}}{c_{11} + c_{21}} \tag{34c}$$

$$\eta_4 = \frac{c_{11} + c_{22}}{c_{11} + c_{12} + c_{21} + c_{22}} \tag{34d}$$

η_5 is F1 score.

$$\eta_5 = 2 \times \frac{\eta_3 \times \eta_1}{\eta_3 + \eta_1} = \frac{2 \times c_{11}}{2 \times c_{11} + c_{12} + c_{21}} \tag{35}$$

and η_6 is Matthews correlation coefficient (MCC)

$$\eta_6 = \frac{c_{11} \times c_{22} - c_{21} \times c_{12}}{\sqrt{(c_{11} + c_{21}) \times (c_{11} + c_{12}) \times (c_{22} + c_{21}) \times (c_{22} + c_{12})}}. \tag{36}$$

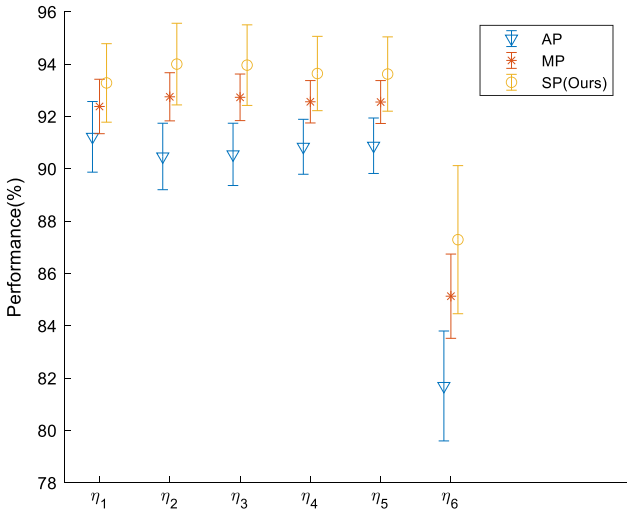


Fig. 9 Error bar of different pooling methods

The mean and standard deviation (SD) of all six measures $\vec{\eta}$ will be calculated over all ten runs.

$$\text{mean}(\eta_m) = \frac{1}{10} \times \sum_{r=1}^{10} \eta_m(r) \tag{37a}$$

$$\text{SD}(\eta_m) = \sqrt{\frac{1}{9} \times \sum_{r=1}^{10} |\eta_m(r) - \text{mean}(\eta_m)|^2}, \tag{37b}$$

where $1 \leq m \leq 6$ represents the index of measures.

4 Experiments, results, and discussion

4.1 Pooling method comparison

The results of 10 runs $\vec{\eta}$ of SP were compared against AP and MP. We compared three pooling methods on test set. The results of all three pooling methods are listed in Table 5. For AP, it obtains $\eta_1 = 91.22 \pm 1.35$, $\eta_2 = 90.47 \pm 1.27$, $\eta_3 = 90.55 \pm 1.19$, $\eta_4 = 90.84 \pm 1.05$, $\eta_5 = 90.88 \pm 1.06$, and $\eta_6 = 81.70 \pm 2.10$. The results of AP are the worst of all three pooling methods. MP obtains better results than AP. The six measures of MP are $\eta_1 = 92.38 \pm 1.04$, $\eta_2 = 92.75 \pm 0.92$, $\eta_3 = 92.73 \pm 0.89$., $\eta_4 = 92.56 \pm 0.81$, $\eta_5 = 92.55 \pm 0.82$, and $\eta_6 = 85.13 \pm 1.61$. Finally, SP obtains the greatest performances on all six measures. The six measures of SP are as follows: $\eta_1 = 93.28 \pm 1.50$, $\eta_2 = 94.00 \pm 1.56$, $\eta_3 = 93.96 \pm 1.54$, $\eta_4 = 93.64 \pm 1.42$, $\eta_5 = 93.62 \pm 1.42$, and $\eta_6 = 87.29 \pm 2.83$. For the ease of clear view, Fig. 9 presents the error bar plot of comparison of all three pooling methods.

Table 6 SC setting

SC	γ_{CB}	γ_{FCB}
I	2	1
II	2	2
III	2	3
IV	3	1
V (Ours)	3	2
VI	3	3

SC structure configuration, γ_{CB} number of CBs, γ_{FCB} number of FCBs

4.2 Structure comparison

We set the number of CBs as γ_{CB} and the number of FCB as γ_{FCB}. We set $\gamma_{CB} = 3$ and $\gamma_{FCB} = 2$ by trial-and-error method. Suppose we all use SP, and we create five different structure configurations (SC) setting as in Table 6. The results of cognate performances on test set $\vec{\eta}$ are shown in Table 7, where we can observe the SC-V performs the best results, which corresponds to our optimal SC setting: $\gamma_{CB} = 3$ and $\gamma_{FCB} = 2$

4.3 Comparison to State-of-the-art approaches

We compare our method "5L-DCNN-SP-C" with other COVID-19 classification approaches: RBFNN [6], K-ELM [7], ELM-BA [8], 6L-CNN-F [9], GoogLeNet [10], ResNet-18 [11]. The results $\vec{\eta}$ on ten runs over test set are presented in Table 8. It is easily observed that our proposed 5L-DCNN-SP-C smashes all the other six comparison baseline methods in all indicators. Particularly, 6L-CNN-F [9] also used convolutional neural network method, and they used more layers (6 layers) than layers used in our model (5 layers).

The reason why our five-layer model is better than that six-layer model [9] is threefold: (i) We choose SP to improve the performance of our deep learning model; (ii) We fine-tune the hyperparameters (such as γ_{CB}, γ_{FCB}, number of filters at each CB, number of neurons at each FCB); (iii) Our model was particularly designed for detecting COVID-19, while the 6L-CNN-F [9] was designed for fingerspelling recognition. In the future, we shall try to use clustering techniques [28, 29] to help improve the performance. Figure 10 shows the comparison bar plot of all seven methods.

5 Conclusion

This study proposed a novel 5L-DCNN-SP-C framework, that combines deep convolutional neural network and stochastic pooling for COVID-19 diagnosis. We added batch normalization transform and dropout layers, and proposed two new blocks (convolution block and fully connected

Table 7 Performances of all six SCs (bold means the best)

SC	η_1	η_2	η_3	η_4	η_5	η_6
I	92.28 ± 0.69	91.00 ± 1.53	91.13 ± 1.39	91.64 ± 0.86	91.70 ± 0.82	83.30 ± 1.72
II	93.13 ± 1.32	92.59 ± 1.36	92.65 ± 1.26	92.86 ± 1.00	92.88 ± 1.00	85.73 ± 1.99
III	93.28 ± 0.61	92.97 ± 1.33	93.01 ± 1.23	93.13 ± 0.71	93.14 ± 0.68	86.26 ± 1.43
IV	92.69 ± 1.11	92.53 ± 1.88	92.57 ± 1.71	92.61 ± 1.08	92.62 ± 1.04	85.24 ± 2.14
V (Ours)	93.28 ± 1.50	**94.00 ± 1.56**	**93.96 ± 1.54**	**93.64 ± 1.42**	**93.62 ± 1.42**	**87.29 ± 2.83**
VI	**93.44 ± 1.52**	93.03 ± 1.09	93.07 ± 0.98	93.23 ± 0.82	93.24 ± 0.85	86.49 ± 1.65

Table 8 Comparison with SOTA approaches (Unit: %)

Approach	η_1	η_2	η_3	η_4	η_5	η_6
RBFNN [6]	67.08	74.48	72.52	70.78	69.64	41.74
K-ELM [7]	57.29	61.46	59.83	59.38	58.46	18.81
ELM-BA [8]	57.08 ± 3.86	72.40 ± 3.03	67.48 ± 1.65	64.74 ± 1.26	61.75 ± 2.24	29.90 ± 2.45
6L-CNN-F [9]	81.04 ± 2.90	79.27 ± 2.21	79.70 ± 1.27	80.16 ± 0.85	80.31 ± 1.13	60.42 ± 1.73
GoogLeNet [10]	76.88 ± 3.92	83.96 ± 2.29	82.84 ± 1.58	80.42 ± 1.40	79.65 ± 1.92	61.10 ± 2.62
ResNet-18 [11]	78.96 ± 2.90	89.48 ± 1.64	88.30 ± 1.50	84.22 ± 1.23	83.31 ± 1.53	68.89 ± 2.33
5L-DCNN-SP-C (Ours)	93.28 ± 1.50	94.00 ± 1.56	93.96 ± 1.54	93.64 ± 1.42	93.62 ± 1.42	87.29 ± 2.83

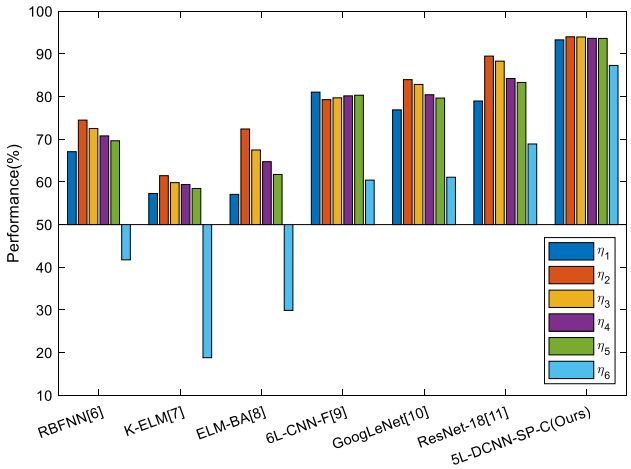


Fig. 10 Comparison to state-of-the-art approaches

block). In our test, we proved three CBs and two FCBs structure can give the best performance.

There are several shortcomings of our method: (i) The dataset is somewhat small. We shall seek to collect more datasets. (ii) Some new network technologies would be tried in our future studies, such as the recent transfer learning pre-trained models.

Acknowledgement This paper is partially supported by Natural Science Foundation of China (61602250); Henan Key Research and Development Project (182102310629); Guangxi Key Laboratory of Trusted Software (kx201901); 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); and British Heart Foundation Accelerator Award, UK.

References

1. Hadi, A.G., Kadhom, M., Hairunisa, N., Yousif, E., Mohammed, S.A.: A review on COVID-19: origin, spread, symptoms, treatment, and prevention. Biointerface Res. Appl. Chem. **10**, 7234–7242 (2020)
2. Tsuchida, T., Fujitani, S., Yamasaki, Y., Kunishima, H., Matsuda, T.: Development of a protective device for RT-PCR testing SARS-CoV-2 in COVID-19 patients. Infect. Control Hosp. Epidemiol. **41**, 975–976 (2020)
3. Penarrubia, L., Ruiz, M., Porco, R., Rao, S.N., Juanola-Falgarona, M., Manissero, D., et al.: Multiple assays in a real-time RT-PCR SARS-CoV-2 panel can mitigate the risk of loss of sensitivity by new genomic variants during the COVID-19 outbreak. Int. J. Infect. Dis. **97**, 225–229 (2020)
4. Ai, T., Yang, Z., Hou, H., Zhan, C., Chen, C., Lv, W. et al.: Correlation of chest CT and RT-PCR testing in coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases. Radiology, Article ID: 200642, (2020)
5. Li, Y., Xia, L.: Coronavirus disease 2019 (COVID-19): role of chest CT in diagnosis and management. AJR Am. J. Roentgenol. **214**, 1280–1286 (2020)
6. Lu, Z.: A pathological brain detection system based on radial basis function neural network. J. Med. Imaging Health Inform. **6**, 1218–1222 (2016)
7. Yang, J.: A pathological brain detection system based on kernel based ELM. Multimed. Tools Appl. **77**, 3715–3728 (2018)
8. Lu, S.: A pathological brain detection system based on extreme learning machine optimized by bat algorithm. CNS Neurol. Disord. Drug. Targets **16**, 23–29 (2017)
9. Jiang, X.: Chinese sign language fingerspelling recognition via six-layer convolutional neural network with leaky rectified linear units

for therapy and rehabilitation. J. Med. Imaging Health Inform. **9**, 2031–2038 (2019)
10. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D. et al.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
11. Yu, X., Wang, S.-H.: Abnormality diagnosis in mammograms by transfer learning based on ResNet18. Fundam. Inform. **168**, 219–230 (2019)
12. Albahri, A.S., Hamid, R.A., Alwan, J.K., Al-qays, Z.T., Zaidan, A.A., Zaidan, B.B. et al.: Role of biological data mining and machine learning techniques in detecting and diagnosing the novel coronavirus (COVID-19): a systematic review. J. Med. Syst. **44**, 11, Article ID: 122, (2020).
13. De Felice, F., Polimeni, A.: Coronavirus Disease (COVID-19): A Machine Learning Bibliometric Analysis. In Vivo **34**, 1613–1617 (2020)
14. Liu, A.J.: Tea category identification using computer vision and generalized eigenvalue proximal SVM. Fundam. Inform. **151**, 325–339 (2017)
15. Zhan, T.M., Chen, Y.: Multiple sclerosis detection based on biorthogonal wavelet transform, RBF kernel principal component analysis, and logistic regression. IEEE Access **4**, 7567–7576 (2016)
16. Du, S.: Multi-objective path finding in stochastic networks using a biogeography-based optimization method. Simulation **92**, 637–647 (2016)
17. Atangana, A.: Application of stationary wavelet entropy in pathological brain detection. Multimed. Tools Appl. **77**, 3701–3714 (2018)
18. Pan, H., Zhang, C., Tian, Y.: RGB-D image-based detection of stairs, pedestrian crosswalks and traffic signs. J. Vis. Commun. Image Represent. **25**, 263–272 (2014)
19. Cui, N., Zou, X.G., Xu, L.: Preliminary CT findings of coronavirus disease 2019 (COVID-19). Clin. Imaging **65**, 124–132 (2020)
20. Tuncer, T., Dogan, S., Ozyurt, F.: An automated residual exemplar local binary pattern and iterative ReliefF based COVID-19 detection method using chest X-ray image. Chemom. Intell. Lab. Syst. **203**, 11, Article ID: 104054, (2020)
21. Hong, J.: Classification of cerebral microbleeds based on fully-optimized convolutional neural network. Multimed. Tools Appl. **79**, 15151–15169 (2020)
22. Hong, J.: Sensorineural hearing loss identification via nine-layer convolutional neural network with batch normalization and dropout. Multimed. Tools Appl. **79**, 15135–15150 (2020)
23. Wang, S., Chen, Y.: Fruit category classification via an eight-layer convolutional neural network with parametric rectified linear unit and dropout technique. Multimed. Tools Appl. **79**, 15117–15133 (2020)
24. Sui, Y.X.: Classification of Alzheimer's disease based on eight-layer convolutional neural network with leaky rectified linear unit and max pooling. J. Med. Syst. vol. **42**, Article ID: 85, (2018)
25. Jiang, Y.Y.: Cerebral micro-bleed detection based on the convolution neural network with rank based average pooling. IEEE Access **5**, 16576–16583 (2017)
26. Garbin, C., Zhu, X.Q., Marques, O.: Dropout versus batch normalization: an empirical study of their impact to deep learning. Multimed. Tools Appl. **79**, 12777–12815 (2020)
27. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**, 1929–1958 (2014)
28. Chen, Y., Zhou, L., Pei, S., Yu, Z., Chen, Y., Liu, X., et al.: KNN-BLOCK DBSCAN: fast clustering for large-scale data. IEEE Trans. Syst. Man Cybern. Syst. (2019). https://doi.org/10.1109/TSMC.2019.2956527
29. Chen, Y.W., Hu, X.L., Fan, W.T., Shen, L.L., Zhang, Z., Liu, X. et al.: Fast density peak clustering for large scale data based on kNN. Knowl. Based Syst. **187**, 7. Article ID: 104824

Prof. Yu-Dong Zhang received his Ph.D. degree in Signal and Information Processing from Southeast University in 2010. He worked as a postdoc from 2010 to 2012 with Columbia University, USA, and as an assistant research scientist from 2012 to 2013 with Research Foundation of Mental Hygiene (RFMH), USA. Now he serves as a Professor with Department of Informatics, University of Leicester, UK. His research interests include deep learning and medical image analysis.

Dr. Suresh Chandra Satapathy is Ph.D. in Computer Science Engineering, currently working as Professor of School of Computer Engg and Dean—Research at KIIT (Deemed to be University), Bhubaneshwar, Odisha, India. He has developed two new optimization algorithms known as Social Group Optimization (SGO) published in Springer Journal and SELO (Social Evolution and Learning Algorithm) published in Elsevier. He has more than 150 publications in reputed journals and conf proceedings.

Dr. Shuaiqi Liu received his Ph.D. degree in Institute of Information Science from Beijing Jiaotong University in 2014 and got B.S. degree in the Department of Information and Computer Science from Shandong University of Science and Technology in 2009. At present, he is an associate professor in College of Electronic and Information Engineering, Hebei University. And he is a visiting scholar at Ottawa University from August 2016 to January 2017. His research interests include image processing and signal processing.

Professor Guang-Run Li obtained a bachelor's degree from Nanjing Medical University in 2005. He is currently the director of the Department of Imaging of Jinhu People's Hospital. He has presided over several municipal scientific research projects and published more than 20 papers. His main research interests are imaging technology and diagnosis, artificial intelligence, including the application of radiology and deep learning in imaging.