A Bagging Dynamic Deep Learning Network for Diagnosing COVID-19

Zhijun Zhang (✉ auzjzhang@scut.edu.cn)

School of Automation Science and Engineering, South China University of Technology, Guangzhou, China

Bozhao Chen

School of Automation Science and Engineering, South China University of Technology, Guangzhou, China

Jiansheng Sun

School of Automation Science and Engineering, South China University of Technology, Guangzhou, China

Research Article

Keywords: COVID-19, X-ray chest radiography image, bagging dynamic deep learning network (B-DDLN)

Posted Date: April 15th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-412040/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Version of Record: A version of this preprint was published at Scientific Reports on August 11th, 2021.
See the published version at <https://doi.org/10.1038/s41598-021-95537-y>.

A Bagging Dynamic Deep Learning Network for Diagnosing COVID-19

Zhijun Zhang^{1,2,3,*}, Bozhao Chen¹, and Jiansheng Sun¹

¹School of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China

²Guangdong Artificial Intelligence and Digital Economy Laboratory (Pazhou Lab), Guangzhou 510335, China

³School of Automation Science and Engineering, East China Jiaotong University, Nanchang 330052, China

*Corresponding author: auzjzhang@scut.edu.cn

ABSTRACT

COVID-19 is a serious epidemic all over the world. As an efficient way in intelligent medical services, using X-ray chest radiography image for automatically diagnosing COVID-19 provides huge assistances and conveniences for clinicians in practice. In this paper, a bagging dynamic deep learning network (B-DDLN) is proposed for diagnosing COVID-19 by intelligently recognizing X-ray chest radiography images. After a series of preprocessing steps for images, we pre-train convolution blocks as a feature extractor. For the extracted features, bagging dynamic learning network classifier is trained based on neural dynamic learning algorithm and bagging algorithm. B-DDLN connects feature extractor and bagging classifier in series. Experimental results verify that using the proposed B-DDLN can achieve 98.8889% testing accuracy, which illustrates the best diagnosis performances among existing methods on the open image set and provides evidences for further detection and treatment.

Introduction

COVID-19 has been rapidly and widely spreading as an ongoing pandemic throughout the world since the end of 2019, which is a major cause of death in humans¹⁻³. According to statistics on Worldometers, nearly 128 million people have been diagnosed as COVID-19 cases and about 2.80 million deaths have occurred up to the end of March, 2021. Most detection methods for COVID-19, including the nucleic acid amplification testing in common, rely on the pathogen testing⁴. However, this test has some limitations. Firstly, it requires testing kits which have limited availability in the supply chain⁵. Secondly, it is time-consuming because the steps of detection are tedious with high technical requirements^{6,7}.

As a vital verification way with simpler operations, recognition of X-ray chest radiography images can be done to screening out part of COVID-19 patients to make diagnosis results and the severity of disease more intuitional^{8,9}. For X-ray chest radiography images, the required equipments can be lightweight and transportable¹⁰. The whole recognizing process takes much less time (about 15 seconds per patient¹⁰) and manual operations to give final visualized results, providing huge assistances and conveniences to clinicians in practice¹¹. Especially in China, many cases were identified as suspected of COVID-19 if characteristic manifestations in X-ray scans were observed^{11,12}. In the field of intelligent medical diagnosis, deep learning has been widely applied in recent years due to its excellent feature extraction and classification abilities¹³. Since the dangers of COVID-19 become severer, increasing number of researchers have applied deep learning and computer vision technologies to diagnose COVID-19 by recognizing X-ray chest radiography images. Zhang et al. proposed the confidence-aware anomaly detection model, which consists of a shared feature extractor, an anomaly detection module, and a confidence prediction module, which achieves AUC of 0.8361 and sensitivity of 0.7170¹⁴. Majeed et al. designed a convolutional neural network (CNN) model for COVID-19 detection from X-ray chest radiography images, which obtains 0.9315 sensitivity and 0.9786 specificity¹⁵. Brunese et al. considered a deep learning network based on the VGG16 model by exploiting transfer learning with the average time for approximately 2.5 seconds and the average 97.00% accuracy¹⁶. On the face of it, the accuracy and related performances are at a great diagnosis level with low misdiagnosis rate. However, from their CNN models we can see that without shortcut connections, it may result in vanishing-gradient problem when training these models.

To overcome this problem, He et al. proposed residual convolutional neural network (ResNet) in 2015, which solves the problem of vanishing gradient effectively by adding shortcut connections and enhancing training efficiency¹⁷⁻²⁰. Inspired by the method in reference¹⁸, Luz et al. applied ResNet50 for COVID-19 patterns detection in X-ray images, which obtains 83.50% diagnosis accuracy²¹. In clinical practice, it is expected to explore the higher accuracy of diagnosing method. Thus, based on the fine-tuned ResNet model, Farooq et al. developed a deep learning framework called COVID-ResNet for screening of COVID-19 from radiographs, which achieves 96.23% accuracy and 100% sensitivity with only 41 epochs²². For further

developments of ResNet, DenseNet was proposed in 2017 by designing dense blocks, which alleviates the vanishing-gradient problem, as well as strengthening feature propagation, encouraging feature reuse, and substantially reducing the number of parameters²³ more effectively. Based on DenseNet, Wang et al. designed and proposed COVID-Net architecture for COVID-19 diagnosis, which achieves 93.33% accuracy by recognizing X-ray images of normal, COVID-19 and ordinary pneumonia cases²⁴. However, the generalization performance of fully-connected layers at the end of ResNet and DenseNet models may not be strong enough to work as a classifier to discriminate and classify deep convolutional features²⁵.

For the classifiers substituting fully-connected layers, support vector machine (SVM) is one of the proper choices to classify samples precisely by searching the optimal decision boundary. Researchers have combined CNN and SVM for image recognition, where CNN is applied for feature extraction and SVM is for classification for these deep extracted features. In diagnosis of COVID-19, Sethy et al. used SVM to classify deep image features extracted from the fully-connected layer of ResNet50, which obtains 95.38% diagnosis accuracy²⁶. To enhance the classification efficiency, Novitasari et al. removed the fully-connected layers of the pre-trained ResNet18 as a feature extractor. For the image features extracted from the global average pooling layer, they utilized principal component analysis and relief methods in feature selection and applied SVM as a classifier, which obtains 96.15% diagnosis accuracy²⁷. However, for the nonlinear distribution of samples, we should find an appropriate kernel function to map these samples into linear separated space, which occupies too much computational sources and model training time.

In order to improve the training efficiency of models under the restrictions of high real-time requirement and limited hardware resources, neural dynamic learning algorithm (NDLA) has been universally adopted due to its characteristics of error exponential convergence of errors and parallel computing. For example, in order to solve time varying convex quadratic programming problem constrained by linear equality and obtain online solution to the time varying Sylvester equation, a varying-parameter convergent-differential neural network (VP-CDNN) is proposed by using an exponential-type time varying design function^{28,29}. Related theoretical analysis has proved that using VP-CDNN can make the residual error over the continuous time converge to zero super-exponentially. Based on the proposed VP-CDNN, various practical problems about time varying over-determined system³⁰, time varying complex Sylvester equation³¹, disturbed time varying inversion systems³², robot tracking^{33,34} and unmanned aerial vehicle controller³⁵ have been effectively solved. Furthermore, Zhang et al. designed an adaptive multi-layer neural dynamics-based controllers of multi-rotor unmanned aerial vehicles for the sake of adapting to various complex transportation³⁶. In addition, to solve time varying quadratic programming problem and robot tracking problem, a power-type varying-parameter recurrent neural network was proposed different from VP-CDNN^{37,38}. To solve non-repetitive motion problem of redundant robot manipulators, Zhang et al. proposed an adaptive fuzzy recurrent neural network, which can avoid the saturation of time varying design functions³⁹. For the application in machine learning and pattern recognition, voting convergent difference neural network was proposed for diagnosing the presence of breast cancer and breast tumor types, which obtains 100% diagnosis accuracy⁴⁰. Thus, NDLA has faster convergent rate and higher accuracy³⁰, and the network training by NDLA is more suitable for the problems of high real-time requirement and limited hardware resources. However, to the best of authors' knowledge, NDLA-based network have not been applied in the field of deep learning and image classification.

In this paper, a bagging dynamic deep learning network (B-DDLN) is proposed for diagnosing COVID-19, which consists of two modules, i.e., feature extractor and bagging classifier. In the proposed B-DDLN, 5 pre-trained convolution blocks are used as feature extractor for X-ray chest radiography images, and bagging classifier based on NDLA and bagging algorithm is applied to classify these image features. The structure of bagging classifier is lightweight, and its cost function is more easily designed in comparison to most classifiers. The proposed B-DDLN can rapidly make both the training and testing errors converge and enhance diagnosing precision. Due to the utilization of bagging classifier, B-DDLN can solve class-imbalance problem by random under-sampling, and further enhance the accuracy and reliability of diagnosis results.

Bagging dynamic deep learning network

In this section, the proposed bagging dynamic deep learning network (B-DDLN) is designed and analyzed in details with four parts. First of all, the construction of B-DDLN model is proposed and applied for diagnosing COVID-19. Secondly, for the design of classifier, topology and principle of dynamic learning network are presented. Thirdly, neural dynamic learning algorithm (NDLA) of dynamic learning network is detailedly derived. Fourthly, based on bootstrap aggregating and voting decision method among several dynamic learning networks trained by NDLA with different mapping functions, bagging dynamic learning network classifier is proposed to enhance the generalization of dynamic learning network.

Construction of B-DDLN diagnosis model

For the application of medical image classification, convolutional neural network is very effective due to its convolution blocks for extracting differentiated image features^{17,18}. However, the generalization performance of fully-connected layers in convolutional neural network may not be strong enough to work as a classifier to discriminate and classify deep convolutional

Figure 1. Construction of the proposed B-DDLN diagnosis model. Pre-trained convolution blocks are used for extracting features of X-ray chest radiography images, and bagging dynamic learning network classifier is responsible for recognizing these features to generate final diagnosis results.

features²⁵. Especially for diagnosing COVID-19, the accuracy and precision are unable to be stable at our satisfied level when simply using convolutional neural network. To enhance generalization performance of diagnosis model, B-DDLN is proposed in this paper, which consists of two modules, i.e., feature extractor and bagging classifier. Pre-trained convolution blocks are used as feature extractor for X-ray chest radiography images, and bagging dynamic learning network classifier based on NDLA and bagging algorithm is applied to classify these image features to generate diagnostic results.

Construction of the proposed B-DDLN is shown in Fig. 1. In Fig. 1, we use images from training set to pre-train 5 designed convolution blocks as a feature extractor. Suppose a three-channel image tensor $\mathbf{g}_s \in \mathbb{R}^{224 \times 224 \times 3}$ is input into this feature extractor, its corresponding feature vector $h(\mathbf{g}_s) \in \mathbb{R}^{1 \times 512}$ can be obtained. After gathering all the features of training images from the feature extractor, we use these features to train bagging dynamic learning network classifier and evaluate the whole proposed B-DDLN.

Since the proposed B-DDLN diagnosis model combines convolution blocks as a feature extractor and bagging dynamic learning network classifier, the constructed principle is termed as B-DDLN constructed algorithm. To better understand it, the detailed steps are illustrated in Algorithm 1.

Algorithm 1 B-DDLN constructed algorithm

Input: Images with actual labels $\{\mathbf{G}, \mathbf{T}\} = \{\mathbf{g}_s, \mathbf{t}_s; 1 \leq s \leq l, s \in \mathbb{Z}\}$, where $\mathbf{G}$ stores X-ray chest radiography images and $\mathbf{T}$ stores labels.

Output: B-DDLN diagnosis model, including feature extractor $h(\mathbf{G})$ and bagging classifier.

- 1: Divide images into training ones $\{\mathbf{G}_{\text{train}}, \mathbf{T}_{\text{train}}\}$ and testing ones $\{\mathbf{G}_{\text{test}}, \mathbf{T}_{\text{test}}\}$.
 - 2: Use the training images $\{\mathbf{G}_{\text{train}}, \mathbf{T}_{\text{train}}\}$ to pre-train the convolution blocks as a feature extractor.
 - 3: Use the pre-trained feature extractor to obtain corresponding features of training images $\{h(\mathbf{G}_{\text{train}}), \mathbf{T}_{\text{train}}\}$.
 - 4: Randomly collect part of the extracted features among all for N times to construct N subsets. Use these subsets to train N dynamic learning networks by NDLA with different mapping functions and then form bagging dynamic learning network classifier.
 - 5: Connect the pre-trained feature extractor and bagging dynamic learning network classifier in series as B-DDLN.
 - 6: **return** B-DDLN diagnosis model.
-

Topology and principle of dynamic learning network

In the bagging classifier module of the proposed B-DDLN, a three-layer dynamic learning network is designed and proposed. Some variables are explained as below, i.e.,

l : Amount of input samples.

m : Feature dimensions.

n : Amount of hidden neurons.

q : Amount of classes for input samples.

$\mathbf{X}$: An $l \times m$ input matrix storing l samples with m dimensions. $x_{si} (1 \leq s \leq l, 1 \leq i \leq m; s, i \in \mathbb{Z}) \in \mathbf{X}$ represents the value of the i th dimension for the s th sample.

$\mathbf{V}$: An $m \times n$ matrix storing weights connecting input and hidden layers. $v_{ij}(1 \leq i \leq m, 1 \leq j \leq n; i, j \in \mathbb{Z}) \in \mathbf{V}$ represents the weight component connecting the i th input neuron with the j th hidden neuron.

$\mathbf{I}$: An $l \times n$ matrix input into the hidden layer.

$\mathbf{Q}$: An $l \times n$ hidden output matrix correspond to $\mathbf{I}$.

$\mathbf{W}$: An $n \times q$ matrix storing weights connecting hidden and output layers. $w_{jr}(1 \leq j \leq n, 1 \leq r \leq q; j, r \in \mathbb{Z}) \in \mathbf{W}$ represents the weight component connecting the j th hidden neuron with the r th output neuron.

$\mathbf{Y}$: An $l \times q$ predicted diagnosis output matrix.

$f_j(\cdot)(1 \leq j \leq n; j \in \mathbb{Z})$: Activation functions in the j th hidden neurons.

$g(\cdot)$: Activation function in the output neurons.

$\bar{\mathbf{Y}}, \mathbf{L}$: $l \times q$ label matrices, where $\bar{\mathbf{Y}}$ is encoded as $-1-1$ format and $\mathbf{L}$ is encoded by one-hot vectors.

$\mathbf{P}$: An $l \times q$ class possibility matrix calculated by softmax formula.

ε : Training error of dynamic learning network calculated by cross entropy formula.

The feed-forward output of dynamic learning network is formulated as

$$\mathbf{Y} = g(\mathbf{QW}) = g(\mathbf{F}(\mathbf{I})\mathbf{W}) \in \mathbb{R}^{l \times q} \quad (1)$$

where

$$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{l1} & \cdots & x_{lm} \end{bmatrix} \in \mathbb{R}^{l \times m} \quad (2)$$

$$\mathbf{Q} = \mathbf{F}(\mathbf{I}) = \mathbf{F}(\mathbf{XV}) \in \mathbb{R}^{l \times n} \quad (3)$$

$$\mathbf{F}(\mathbf{I}) = [f_1(\mathbf{i}_1); f_2(\mathbf{i}_2); f_3(\mathbf{i}_3); \cdots; f_j(\mathbf{i}_j); \cdots; f_n(\mathbf{i}_n)] \quad (4)$$

$$\mathbf{W} = \begin{bmatrix} w_{11} & \cdots & w_{1q} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nq} \end{bmatrix} \in \mathbb{R}^{n \times q}. \quad (5)$$

In equation (4), $\mathbf{i}_j \in \mathbb{R}^{l \times 1}$ denotes the j th column vector of $\mathbf{I} \in \mathbb{R}^{l \times n}$. For activation functions in the hidden and output neurons, two cases are listed, i.e.,

Case 1: All the activation functions are set as softsign function. Its expression is

$$f_j(z) = g(z) = \frac{z}{1 + |z|} (1 \leq j \leq n; j \in \mathbb{Z}). \quad (6)$$

Case 2: Activation functions of the output neurons are still set as softsign function, while those of the hidden neurons are power-softsign function. The expression of power-softsign function is

$$f_j(z) = \left(\frac{z}{1 + |z|}\right)^{j-1} (1 \leq j \leq n; j \in \mathbb{Z}). \quad (7)$$

As for the s th sample $\mathbf{x}_s = [x_{s1}, \cdots, x_{sm}] \in \mathbb{R}^{1 \times m}$ ($1 \leq s \leq l; s \in \mathbb{Z}$) from $\mathbf{X}$, the output through dynamic learning network $\mathbf{y}_s = [y_{s1}, \cdots, y_{sq}] \in \mathbb{R}^{1 \times q}$ is calculated by equation (1), and corresponding class probabilities vector $\mathbf{P}_s$ is obtained by softmax formula

$$\mathbf{P}_s = [P_{s1}, \cdots, P_{sq}] = \left[\frac{\exp(y_{s1})}{\sum_{r=1}^q \exp(y_{sr})}, \cdots, \frac{\exp(y_{sq})}{\sum_{r=1}^q \exp(y_{sr})} \right] \in \mathbb{R}^{1 \times q} \quad (8)$$

where $P_{s1}, \cdots, P_{sq} \in [0, 1]$, $\sum_{r=1}^q P_{sr} = 1$. If $P_{sr} = \max \{P_{s1}, \cdots, P_{sq}\}$ ($1 \leq r \leq q; r \in \mathbb{Z}$), $\mathbf{x}_s$ is predicted as belonging to the r th class according to Bayesian decision principle based on minimum classification error probability⁴¹. Similarly, the class probability matrix $\mathbf{P}$ for l input samples is calculated below.

$$\mathbf{P} = \begin{bmatrix} P_{11} & \cdots & P_{1q} \\ \vdots & \ddots & \vdots \\ P_{l1} & \cdots & P_{lq} \end{bmatrix} = \begin{bmatrix} \frac{\exp(y_{11})}{\sum_{r=1}^q \exp(y_{1r})} & \cdots & \frac{\exp(y_{1q})}{\sum_{r=1}^q \exp(y_{1r})} \\ \vdots & \ddots & \vdots \\ \frac{\exp(y_{l1})}{\sum_{r=1}^q \exp(y_{lr})} & \cdots & \frac{\exp(y_{lq})}{\sum_{r=1}^q \exp(y_{lr})} \end{bmatrix} \in \mathbb{R}^{l \times q}. \quad (9)$$

Figure 2. Diagram of neural dynamic learning algorithm. If training error $\varepsilon(k)$ satisfies $\varepsilon(k) < \varepsilon'$, where ε' denotes a threshold, $W(k)$ will be updated as $W(k+1)$ at the $(k+1)$ th training round.

Learning algorithm of dynamic learning network

In recent years, neural dynamic learning algorithm (NDLA) has been favored because of more rapid convergence and higher precision^{37,42}, whose design formula is expressed as

$$\dot{e}(t) = -\lambda \Phi(e(t)). \quad (10)$$

In equation (10), $t \in \mathbb{R}$ denotes continuous time. $e(t) \in \mathbb{R}$ denotes deviation between prediction output and expectation over t . $\dot{e}(t)$ denotes the derivative of the deviation $e(t)$ with respect to t . $\lambda > 0, \lambda \in \mathbb{R}$ denotes NDLA parameter. $\Phi(\cdot)$ denotes mapping function, which is a monotonically increasing and odd function^{35,37,40,42}. Because NDLA is conducted in digital computer, such serial time expression is transformed into a discrete ones combined with Euler discrete formula⁴³, which is shown as

$$\frac{e(k+1) - e(k)}{\tilde{h}} = -\lambda \Phi(e(k)) \quad (11)$$

where $\tilde{h} > 0, \tilde{h} \in \mathbb{R}$ denotes the discrete step, and $k > 0, k \in \mathbb{Z}$ denotes discrete time and the k th training round. Equation (11) is further transformed as

$$e(k+1) = e(k) - \alpha \Phi(e(k)) \in \mathbb{R} \quad (12)$$

where $\alpha = \tilde{h}\lambda > 0, \alpha \in \mathbb{R}$ denotes NDLA design coefficient. Thus, equation (12) is expressed as a discrete NDLA design formula.

Theorem 1. A deviation based learning rule is formulated as a discrete NDLA design formula, i.e., $e(k+1) - e(k) = -\alpha \Phi(e(k))$, where $\alpha > 0$ is a proper value within a range. If the mapping function $\Phi(\cdot)$ in NDLA is a monotonically increasing and odd function, then the deviation between prediction output and expectation $e(k)$ shows absolute convergence to zero along with discrete time k increasing.

Proof. The discrete NDLA design formula is expressed as

$$e(k+1) - e(k) = -\alpha \Phi(e(k)). \quad (13)$$

If the mapping function $\Phi(\cdot)$ is a monotonically increasing and odd function, $\Phi(e(k)) > 0$ is obtained when the deviation $e(k)$ satisfies $e(k) > 0$. Because NDLA coefficient satisfies $\alpha > 0$, $-\alpha \Phi(e(k)) < 0$ is then deduced, which leads to $e(k+1) < e(k)$ according to equation (13). Conversely, when the deviation $e(k) < 0$, $-\alpha \Phi(e(k)) > 0$ is obtained, and $e(k+1) > e(k)$ can be deduced further. Thus, $|e(k+1)| < |e(k)|$ is deduced. With the discrete time k increasing, the absolute value of the deviation will absolutely converge to zero, which is expressed as $\lim_{k \rightarrow +\infty} |e(k)| = 0$. The proof is completed. $\square$

For the mapping function, the linear, tanh and sinh types of mapping functions are utilized in this paper, i.e.,

$$\Phi_1(z) = z \quad (14)$$

$$\Phi_2(z) = \tanh(z) \quad (15)$$

$$\Phi_3(z) = \sinh(z). \quad (16)$$

The discrete NDLA design formula in this paper is utilized to train dynamic learning network, where the weight matrix $\mathbf{W}$ is iterated. According to Fig. 2, when the k th training round is finished, the diagnosis deviation $\mathbf{E}(k)$ is calculated by

$$\mathbf{E}(k) = \mathbf{Y}(k) - \bar{\mathbf{Y}} \in \mathbb{R}^{l \times q} \quad (17)$$

where $\mathbf{Y}(k)$ denotes output through dynamic learning network and $\bar{\mathbf{Y}}$ denotes labels of samples in $-1-1$ format. Before the next round, the training error $\varepsilon(k)$ is calculated by cross entropy formula at first, i.e.,

$$\varepsilon(k) = \frac{1}{l} \sum_{s=1}^l \sum_{r=1}^q (-L_{sr} \log(P_{sr})) \in \mathbb{R} \quad (18)$$

where $L_{sr} \in \mathbf{L}$ and $P_{sr} \in \mathbf{P}(k)$. The class probability matrix $\mathbf{P}(k)$ is calculated according to equation (9). Suppose the threshold of training error is $\varepsilon' \in \mathbb{R}$. If $\varepsilon(k) < \varepsilon'$, stop and quit this training process. Otherwise, continue updating $\mathbf{W}$ to train dynamic learning network.

For training details of dynamic learning network in the $(k+1)$ th training round, the diagnosis deviation of $\mathbf{E}(k+1)$ is obtained by the following equation, i.e.,

$$\mathbf{E}(k+1) = \mathbf{E}(k) - \alpha \Phi(\mathbf{E}(k)) \in \mathbb{R}^{l \times q} \quad (19)$$

where

$$\mathbf{E}(k+1) = \mathbf{Y}(k+1) - \bar{\mathbf{Y}} \in \mathbb{R}^{l \times q}. \quad (20)$$

When $\Phi(\cdot)$ is a monotonically increasing and odd function, $\mathbf{E}(k)$ can rapidly converge to $\mathbf{0}$ along with k increasing according to Theorem 1, which indicates that $\lim_{k \rightarrow +\infty} \mathbf{Y}(k) = \bar{\mathbf{Y}}$ according to equation (17). Through softmax formula (i.e., equation (9)) and encoder, $\lim_{k \rightarrow +\infty} \mathbf{P}(k) = \bar{\mathbf{P}}$ is obtained, where $\bar{\mathbf{P}}$ denotes a constant targeted class probability matrix. Thus, the training error $\varepsilon(k)$ satisfies $\lim_{k \rightarrow +\infty} \varepsilon(k) = \tilde{\varepsilon}$ according to equation (18), which indicates that $\varepsilon(k)$ theoretically converges to a targeted constant $\tilde{\varepsilon}$ along with k increasing.

Substituting equations (17) and (20) to equation (19), we can obtain discrete neural dynamic equation as

$$\mathbf{Y}(k+1) = \mathbf{Y}(k) - \alpha \Phi(\mathbf{E}(k)) \in \mathbb{R}^{l \times q} \quad (21)$$

where

$$\mathbf{Y}(k) = g(\mathbf{QW}(k)) \in \mathbb{R}^{l \times q} \quad (22)$$

$$\mathbf{Y}(k+1) = g(\mathbf{QW}(k+1)) \in \mathbb{R}^{l \times q}. \quad (23)$$

Therefore, $\mathbf{W}(k+1)$ can be figured out under the situation of known $\mathbf{Q}$ and $\mathbf{W}(k)$. Substituting equations (22) and (23) to equation (21), we can obtain

$$g(\mathbf{QW}(k+1)) = g(\mathbf{QW}(k)) - \alpha \Phi(\mathbf{E}(k)) \in \mathbb{R}^{l \times q}. \quad (24)$$

$\mathbf{W}(k+1)$ is figured out by solving equation (24) as

$$\mathbf{W}(k+1) = \mathbf{Q}^+ g^{-1}(g(\mathbf{QW}(k)) - \alpha \Phi(\mathbf{E}(k))) \in \mathbb{R}^{n \times q} \quad (25)$$

which shows the iterated relation between $\mathbf{W}(k+1)$ and $\mathbf{W}(k)$. A training round is thus finished. Without loss of generality, the softsign function is applied as an example of $g(\cdot)$. However, the inverse function of softsign function can not be found. Under this circumstance, an approximate expression $\hat{g}(\cdot)$ is used instead, i.e.,

$$g^{-1}(z) = \hat{g}(z) = \frac{z}{1 - |z|}. \quad (26)$$

Figure 3. Concrete schematic of bagging dynamic learning network classifier in Fig. 1. Several dynamic learning networks trained by different subsets randomly under-sampled from training features predict the belonging class of testing images, respectively at first, and the final diagnosis result based on these predictions is determined by the majority rule.

Bagging dynamic learning network classifier

To further enhance the generalization of dynamic learning network, bootstrap aggregating (bagging) algorithm among is utilized to construct a more robust bagging dynamic learning network classifier, which is suggested in Fig. 1. In bagging algorithm, part of the samples are randomly collected from the entire training features as a subset, which is used for training a dynamic learning network. Such process is repeated for several times, and several trained dynamic learning networks are obtained. In combination strategy, these trained dynamic learning network classifiers predict testing samples, respectively at first. Then in regard to these predicted results, the final belonging class is determined according to the principle of plurality voting, i.e., the majority rule^{40,44}. Specifically, we initialize a zero vector $\mathbf{a}_t = [a_{t1}, \dots, a_{tq}] \in \mathbb{R}^{1 \times q}$ for votes statistic. If the $\hat{k}$ th dynamic learning network model determine that $\mathbf{x}_t$ belongs to the $\hat{r}$ th class ($\hat{k} \in [1, N], \hat{r} \in [1, q]; \hat{k}, \hat{r} \in \mathbb{Z}$), $a_{t\hat{r}}$ will be added by 1. According to the majority rule, we determine the index corresponding to the maximum number of votes among all the elements in $\mathbf{a}_t$ as the final predicted class after votes statistic.

Structure and principle of bagging dynamic learning network classifier are shown in Fig. 3, where dynamic learning networks trained by NDLA with different mapping functions $\Phi(\cdot)$ are simultaneously trained by using different subsets randomly drawn from shuffled training features. Suppose there are 2 kinds of diagnosis results (positive and negative) and N dynamic learning networks. When a testing sample $\mathbf{x}_t = [x_{t1}, \dots, x_{tm}] \in \mathbb{R}^{1 \times m}$ is entered into this proposed bagging dynamic learning network, according to the majority rule, these dynamic learning networks mutually independently judge which kind of diagnosis result it belongs to and then decide the final diagnosis result. For example, when the predicted result of the first dynamic learning network is positive (COVID-19) while those of others are negative (normal), the final diagnosis result is determined as negative. Specially for a practical COVID-19 screening system, if the vote is tied, the final result is directly diagnosed as positive, which can effectively reduce the risk of missed diagnosis by enhancing sensitivity⁴⁰. When the final diagnosis result shows positive, more detailed detections and corresponding treatments need to be handled.

Results and discussions

In this section, the proposed B-DDLN is applied to diagnose presence of COVID-19 by recognizing X-ray chest radiography images. This section consists of five parts, including description and preprocessing of image set, feature extraction, cross validation, experimental results of the proposed B-DDLN, and discussions for comparing B-DDLN with other diagnosis models. The flowchart of the entire experiment is shown in Fig. 4, where the feature extractor and bagging classifier are successively trained by using training images and corresponding features, respectively to construct B-DDLN diagnosis model. Testing images are used for evaluating this proposed B-DDLN diagnosis model, and diagnosis results are obtained.

Figure 4. Flowchart of the entire experiment. Preprocessed X-ray chest radiography images are divided into training and testing sets. Training images are used for pre-training convolution blocks as a feature extractor and training bagging dynamic learning network classifier to construct B-DDLN diagnosis model, and testing images are applied for evaluating the proposed B-DDLN to obtain diagnosis results and performances.

Description and preprocessing of image set

In this paper, we combine and modify the four data repositories to create the COVID-19 image set with binary-classification by leveraging the following types of patient and normal cases from each of the data repositories:

- COVID-19 patient cases from COVID-19 open image data collection, which was created by assembling medical images from websites and publications⁴⁵. This data collection currently contains hundreds of frontal view X-ray images, which is a necessary resource to develop and evaluate tools to aid in the diagnosis of COVID-19⁴⁶. An example is shown in Fig. 5(a).
- COVID-19 patient cases from Figure 1 COVID-19 Chest X-ray Dataset Initiative, which are built to enhance diagnosis models for COVID-19 disease detection and risk stratification. An example is shown in Fig. 5(b).
- COVID-19 patient and normal cases from ActualMed COVID-19 Chest X-ray Dataset Initiative. A COVID-19 example is shown in Fig. 5(c) and a normal example is shown in Fig. 5(e).
- COVID-19 patient and normal cases from COVID-19 Radiography Database (winner of the COVID-19 data set award by Kaggle community), which is created by a team of researchers from Qatar University, Doha, Qatar and the University of Dhaka, Bangladesh along with their collaborators from Pakistan and Malaysia in collaboration with medical doctors. A COVID-19 example is shown in Fig. 5(d) and a normal example is shown in Fig. 5(f).

After gathering X-ray chest radiography images from these collections, there are 2284 images in the newly formed image set (816 from COVID-19 patient and 1468 from normal cases). To improve the quality of image data and enhance image features, preprocessing of X-ray chest radiography images is necessary. First of all, image channel sequence is changed from BGR to RGB. Secondly, the sizes of all the images are uniformly set as $224 \times 224 \times 3$. Thirdly, these images are labeled for training B-DDLN. After the division image set, 180 testing images (100 from normal and 80 from COVID-19 patient cases) among all are selected to evaluate the proposed B-DDLN.

Feature extractor pre-training

For pre-training feature extractor of images, the entire process is carried out in an environment equipped with PyCharm Community Edition of 2020.1.2 version and MATLAB 2020a on a personal computer equipped with an Intel Core i7 processor, NVIDIA GeForce GTX 1660 Ti GPU architecture and 32 GB RAM running under Windows 10 operating system. We adopt scikit-learn, tensorflow 2.2.1 and keras to build convolution blocks and pre-train them. When these convolution blocks finish pre-trained as a feature extractor $h(\cdot)$, a $224 \times 224 \times 3$ preprocessed X-ray chest radiography image tensor $\mathbf{g}_s$ is input into $h(\cdot)$, and its corresponding 512-dimension extracted feature vector $h(\mathbf{g}_s)$ can be obtained as output. Therefore, the training features extracted from training images are utilized to train the bagging dynamic learning network classifier, and the testing features extracted from testing images are used to evaluate this bagging dynamic learning network classifier.

Figure 5. Examples of X-ray chest radiography images collected from COVID-19 patient and normal cases. (A) A COVID-19 patient case from COVID-19 open image data collection. (B) A COVID-19 patient case from Figure 1 COVID-19 Chest X-ray Dataset Initiative. (C) A COVID-19 patient case from ActualMed COVID-19 Chest X-ray Dataset Initiative. (D) A COVID-19 patient case from COVID-19 Radiography Database. (E) A Normal case from ActualMed COVID-19 Chest X-ray Dataset Initiative. (F) A Normal case from COVID-19 Radiography Database.

Random under-sampling of image features

After division of image set, there are 2104 left in the training images, where 736 come from COVID-19 patient and 1368 from normal cases. This distribution causes category imbalance, which hinders the classifier from releasing effective performances. To overcome this problem, random under-sampling of samples (i.e., feature vectors) is necessary to be carried out, which means that 1000 samples from COVID-19 patient cases are randomly selected among all. In this paper, we do this random under-sampling for three times and obtain three different subsets. Each subset consists of 1736 samples, where 736 come from COVID-19 patient and 1000 from normal cases. By using these three subsets, three dynamic learning networks (i.e., $N = 3$) with three different types of mapping functions are trained. As examples, these three types of mapping functions are: linear (i.e., equation (14)), tanh (i.e., equation (15)) and sinh (i.e., equation (16)) types. Furthermore, bagging dynamic learning network classifier is constructed by these three dynamic learning networks and bagging algorithm.

Classification performances of B-DDLN

In this paper, two cases of activation functions are designed in dynamic learning network. In case 1, all the activation functions in the hidden and output neurons are set as softsign function (i.e., equation (6)). In case 2, activation functions in the output neurons also as softsign function while those in the hidden neurons as power-softsign function (i.e., equation (7)).

For classifying the extracted features from convolution blocks of B-DDLN, trends of classification errors calculated according to cross entropy formula (i.e., equation (18)) of dynamic learning networks trained by NDLA in both two cases are shown in Figs. 6 and 7. From Figs. 6 and 7 we can see that both two tendencies are monotonically and rapidly decreasing and

Figure 6. Trends of errors for training and testing dynamic learning network (DLN) in case 1. (A) DLN trained by NDLA with linear type of mapping function. (B) DLN trained by NDLA with tanh type of mapping function. (C) DLN trained by NDLA with sinh type of mapping function.

Figure 7. Trends of errors for training and testing dynamic learning network (DLN) in case 2. (A) DLN trained by NDLA with linear type of mapping function. (B) DLN trained by NDLA with tanh type of mapping function. (C) DLN trained by NDLA with sinh type of mapping function.

Activation Function	Model	Confusion Matrix		Accuracy (%)
Case 1	B-DDLN	TP = 99	FN = 1	98.8889
		FP = 1	TN = 79	
Case 2	B-DDLN	TP = 99	FN = 1	98.8889
		FP = 1	TN = 79	

Table 1. Confusion matrices of the proposed B-DDLN.

even converge to a specific minor value when the training round k is large enough. The testing errors ε_t in all of the subgraphs are always decreasing as k increases. These phenomena indicate that the dynamic learning networks trained by NDLA with different mapping and activation functions are not over-fitting, and verify that using NDLA can rapidly make the classification error monotone convergent and enhance precision.

In the testing process, confusion matrices of the proposed B-DDLNs under both two cases of activation functions are shown in Table 1, where

True (T): Actual result is normal.

False (F): Actual result is COVID-19.

Positive (P): Predicted result is normal.

Negative (N): Predicted result is COVID-19.

Thus, the testing accuracy is calculated by equation (27), i.e.,

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (27)$$

Activation Function	Accuracy and Hyper-parameters	Mapping Function Type		
		Linear	Tanh	Sinh
Case 1	Testing Accuracy (%)	98.3333	97.2222	96.6667
	Validation Accuracy (%)	94.2529	95.4023	96.5517
	Training Accuracy (%)	96.4862	96.4862	96.8318
	Training Time (s)	6.0156	2.3906	1.4688
	Threshold ε'	0.2506	0.3270	0.2947
	Training Rounds k	159	59	31
	NDLA Coefficient α	0.01	0.02	0.04
	Hidden Neurons n	50	55	60
Case 2	Testing Accuracy (%)	98.3333	97.2222	97.7778
	Validation Accuracy (%)	97.7011	98.2759	94.8276
	Training Accuracy (%)	95.5069	96.4862	95.3917
	Training Time (s)	17.5000	2.7344	6.2344
	Threshold ε'	0.2154	0.2711	0.2935
	Training Rounds k	290	40	109
	NDLA Coefficient α	0.01	0.04	0.01
	Hidden Neurons n	42	40	41

Table 2. Accuracies and hyper-parameters of single dynamic learning networks.

Figure 8. ROC curves of dynamic learning networks (DLN) and bagging dynamic learning network (BDLN) in case 1. (A) DLN trained by NDLA with linear type of mapping function. (B) DLN trained by NDLA with tanh type of mapping function. (C) DLN trained by NDLA with sinh type of mapping function. (D) BDLN.

In Table 1, most testing samples distribute in TP and TN areas, which illustrates that the proposed B-DDLNs in two cases obtain high diagnosis accuracy. Accuracies of dynamic learning networks with corresponding hyper-parameters in two cases are listed in Table 2. From Table 2 we can see that the highest testing accuracy of a single dynamic learning network can reach 98.3333%. After plurality voting, the bagging dynamic learning networks in two cases achieve 98.8889% accuracy, which shows great precision for diagnosing COVID-19. The amount of hidden neurons is no more than 60, which suggests that the dynamic learning network is a lightweight network.

In addition, for the testing set, the receiver operating characteristic curves (ROC curves) of the dynamic learning network and its improved model, i.e., bagging dynamic learning network classifiers under two cases are shown in Figs. 8 and 9. It can be seen that these ROC curves approach coordinate point (0, 1), which leads to large corresponding areas under curves (AUC). This phenomena further substantiate great classification performances of the proposed B-DDLN diagnosis model.

Comparison analysis

In order to further verify the superiorities of the proposed B-DDLN diagnosis model, fair and comprehensive comparisons in classification performances between B-DDLN and other existing methods on the COVID-19 image set are carried out in this section.

For comparisons with traditional convolutional neural network (CNN), we regard the whole traditional CNN as a classifier and apply 10-fold cross validation to train it. The number of training parameters and the consumed time for constructing trained models are used to measure their training efficiency. Firstly, due to the same ways in the process of extracting image features, we only compare the number of training parameters in classification. Secondly, the consumed time of constructing

Figure 9. ROC curves of dynamic learning networks (DLN) and bagging dynamic learning network (BDLN) in case 2. **(A)** DLN trained by NDLA with linear type of mapping function. **(B)** DLN trained by NDLA with tanh type of mapping function. **(C)** DLN trained by NDLA with sinh type of mapping function. **(D)** BDLN.

Model	Number of Training Parameters	Training Time (s)	Testing Accuracy (%)
B-DDLN1⁽¹⁾	330	569	98.8889
B-DDLN2⁽²⁾	246	759	98.8889
Traditional CNN	513	2310	97.2222 (Highest)

(1) B-DDLN in case 1.

(2) B-DDLN in case 2.

Table 3. Comparisons with traditional convolutional neural network (CNN) in training efficiency.

Classifier	Testing Accuracy(%)	F1-score	Precision	Specificity	Sensitivity	Kappa Value	AUC
B-DDLN1⁽¹⁾	98.8889	0.9900	0.9900	0.9875	0.9900	0.9775	0.9888
B-DDLN2⁽²⁾	98.8889	0.9900	0.9900	0.9875	0.9900	0.9775	0.9888
BPNN ⁽³⁾	95.5556	0.9588	0.9894	0.9875	0.9300	0.9107	0.9587
Naive Bayes	92.7778	0.9353	0.9307	0.9125	0.9400	0.8536	0.9012
LDA ⁽⁴⁾	93.3333	0.9406	0.9314	0.9125	0.9500	0.8647	0.9313
QDA ⁽⁵⁾	94.4444	0.9485	0.9787	0.9750	0.9200	0.8883	0.9475
Linear SVM ⁽⁶⁾	96.6667	0.9697	0.9796	0.9750	0.9600	0.9327	0.9675
Gaussian SVM	95.5556	0.9600	0.9600	0.9500	0.9600	0.9100	0.9550
<i>k</i> -NN ⁽⁷⁾	95.0000	0.9569	0.9174	0.8875	1.0000	0.8976	0.9437
Logistic Regression	90.5556	0.9171	0.8952	0.8625	0.9400	0.8075	0.9012
Decision Tree	93.3333	0.9394	0.9490	0.9375	0.9300	0.8653	0.9337
Bagging Tree	92.7778	0.9366	0.9143	0.8875	0.9600	0.8528	0.9237
Boosting Tree	96.1111	0.9648	0.9697	0.9625	0.9600	0.9213	0.9612

(1) B-DDLN in case 1.

(2) B-DDLN in case 2.

(3) Back Propagation Neural Network.

(4) Linear Discriminant Analysis.

(5) Quadratic Discriminant Analysis.

(6) Support Vector Machine.

(7) *k*-Nearest Neighbor.

Table 4. Comparison results with other classifiers in evaluation indicators.

trained models includes the processes of 10-fold cross validation and final training. Comparison results are shown in Table 3. From Table 3 it can be seen that testing accuracies of the proposed B-DDLNs in two cases are both higher than that of traditional CNN, which directly indicates higher diagnosis precision of the proposed B-DDLN. Compared with traditional CNN, the total number of training parameters for the proposed B-DDLN saves at least 35.55% in classification, which indicates that B-DDLN is proposed based on several lightweight neural network structures. For the proposed B-DDLN, the consumed time for training model is much shorter than that of traditional CNN, which further reflects that using NDLA can enhance training efficiency of the diagnosis model.

Meanwhile, for the same extracted image features, comparison results in classification performances between the proposed B-DDLN, especially its bagging dynamic learning network and other classifiers are shown in Table 4, where the corresponding experiments are carried out under the same conditions like 10-fold validation. From Table 4 we can see that the specificity and sensitivity of the proposed B-DDLN are 0.9875 and 0.9900, respectively, which indicates that misdiagnosis and missed diagnosis of COVID-19 happen very rarely. Meanwhile, most of the evaluation indicators simultaneously keep the highest among all the classifiers. Such top level substantiates the excellent advantages and superiorities of the proposed B-DDLN in diagnosing the presence of COVID-19.

Conclusion

In this paper, B-DDLN has been proposed for diagnosing presence of COVID-19 by recognizing X-ray chest radiography images. After some preprocessing steps, training images have been used to pre-train convolution blocks as a feature extractor. For the training features extracted from this extractor, based on neural dynamic learning algorithm (NDLA) and bagging algorithm has been trained. Experimental and compared results have verified that the proposed B-DDLN diagnosis model has greatly improved training efficiency and increased diagnosis accuracy. The diagnosis accuracy of the proposed B-DDLN has reached 98.8889%, which is the highest among the existing methods. In addition, corresponding performance indicators of bagging dynamic learning network classifier have reached the top level among all, which has further substantiated that the proposed B-DDLN possesses excellent diagnostic ability. Such accurate diagnosis outcomes provide assertive evidences for early detection of COVID-19, which helps to provide suitable treatments for patients and guarantees human health.

Data availability

COVID-19 image data collection is available at <https://github.com/ieee8023/covid-chestxray-dataset>. Figure 1 COVID-19 chest X-ray dataset initiative is available at <https://github.com/agchung/Figure1-COVID-chestxray-dataset>. Actualmed COVID-19 chest X-ray dataset initiative is available at <https://github.com/agchung/Actualmed-COVID-chestxray-dataset>. COVID-19 radiography database is available at <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database>.

References

1. Zheng, N. et al. Predicting COVID-19 in China using hybrid AI model. *IEEE Trans. Cybern.* **50**, 2891-2904 (2020).
2. Nishio, M., Noguchi, S., Matsuo, H. & Murakami, T. Automatic classification between COVID-19 pneumonia, non-COVID-19 pneumonia, and the healthy on chest X-ray image: combination of data augmentation methods. *Sci. Rep.* **10**, 17532 (2020).
3. Luengo-Oroz, M. et al. Artificial intelligence cooperation to support the global response to COVID-19. *Nat. Mach. Intell.* **2**, 295-297 (2020).
4. Wilson, I. G. Inhibition and facilitation of nucleic acid amplification. *Appl. Environ. Microbiol.* **63**, 3741-3751 (1997).
5. Singh, K. K. & Singh, A. Diagnosis of COVID-19 from chest X-Ray Images using wavelets-based depthwise convolution network. *Big Data Min. Anal.* **4**, 84-93 (2021).
6. Yan, L. et al. An interpretable mortality prediction model for COVID-19 patients. *Nat. Mach. Intell.* **2**, 283-288 (2020).
7. Ai, T. et al. Correlation of chest CT and RT-PCR testing for coronavirus disease 2019 (COVID-19) in China: A report of 1014 cases. *Radiology* **296**, e32-e40 (2020).
8. Korot, E. et al. Code-free deep learning for multi-modality medical image classification. *Nat. Mach. Intell.* <https://doi.org/10.1038/s42256-021-00305-2> (2021).
9. Sauter, A.P. et al. Optimization of tube voltage in X-ray dark-field chest radiography. *Sci. Rep.* **9**, 8699 (2019).
10. Tabik, S. et al. COVIDGR dataset and COVID-SDNet methodology for predicting COVID-19 based on chest X-Ray images. *IEEE J. Biomed. Health Inform.* **24**, 3595-3605 (2020).
11. Shi, F. et al. Review of artificial intelligence techniques in imaging data acquisition, segmentation, and diagnosis for COVID-19. *IEEE Rev. Biomed. Eng.* **14**, 4-15 (2021).
12. Liang, T. *Handbook of COVID-19 Prevention and Treatment*. (The First Affiliated Hospital, School of Medicine, Zhejiang University, China, 2020).
13. Wang, Z. et al. Automatically discriminating and localizing COVID-19 from community-acquired pneumonia on chest X-rays. *Pattern Recognit.* **110**, 107613 (2021).

14. Zhang, J. et al. Viral pneumonia screening on chest X-rays using confidence-aware anomaly detection. *IEEE Trans. Med. Imaging* **40**, 879-890 (2021).
15. Majeed, T., Rashid, R., Ali, D., & Asaad, A. COVID-19 detection using CNN transfer learning from X-ray images. *medRxiv preprint* <https://doi.org/10.1101/2020.05.12.20098954> (2020).
16. Brunese, L., Mercaldo, F., Reginelli, A., & Santone, A. Explainable deep learning for pulmonary disease and coronavirus COVID-19 detection from X-rays. *Comput. Meth. Programs Biomed.* **196**, 105608 (2020).
17. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436-444 (2015).
18. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit. (CVPR)* 770-778 (IEEE, 2016).
19. Bressem, K. K. et al. Comparing different deep learning architectures for classification of chest radiographs. *Sci. Rep.* **10**, 13590 (2020).
20. Dai, X., Wu, X., Wang, B., & Zhang, L. Semisupervised scene classification for remote sensing images: A method based on convolutional neural networks and ensemble learning. *IEEE Geosci. Remote Sens. Lett.* **16**, 869-873 (2019).
21. Luz, E. et al. Towards an effective and efficient deep learning model for COVID-19 patterns detection in X-ray images. Preprint at <https://arxiv.org/pdf/2004.05717v4.pdf> (2020).
22. Farooq, M. & Hafeez, A. COVID-ResNet: A deep learning framework for screening of COVID19 from radiographs. Preprint at <https://arxiv.org/ftp/arxiv/papers/2003/2003.14395.pdf> (2020).
23. Huang, G., Liu, Z., van der Maaten, L. & Weinberger, K. Q. Densely connected convolutional networks. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit. (CVPR)* 2261-2269 (IEEE, 2017).
24. Wang, L., Lin, Z. & Wong, A. COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. *Sci. Rep.* **10**, 19549 (2020).
25. Zeng, Y., Xu, X., Shen, D., Fang Y. & Xiao, Z. Traffic sign recognition using kernel extreme learning machines with deep perceptual features. *IEEE Trans. Intell. Transp. Syst.* **18**, 1647-1653 (2017).
26. Sethy, P. K., Behera, S. K., Ratha, P. K. & Biswas, P. Detection of coronavirus disease (COVID-19) based on deep features and support vector machine. *Int. J. Math. Eng. Manage. Sci.* **5**, 643-651 (2020).
27. Novitasari, D. C. R. et al. Detection of COVID-19 chest X-ray using support vector machine and convolutional neural network. *Commun. Math. Biol. Neurosci.* **2020**, 1-19 (2020).
28. Zhang, Z. et al. Y. A new varying-parameter convergent-differential neural-network for solving time-varying convex QP problem constrained by linear-equality. *IEEE Trans. Autom. Control* **63**, 4110-4125 (2018).
29. Zhang, Z. et al. A new varying-parameter recurrent neural-network for online solution of time-varying Sylvester equation. *IEEE Trans. Cybern.* **48**, 3135-3148 (2018).
30. Zhang, Z., Zheng, L., Qiu, T., & Deng, F. Varying-parameter convergent-differential neural solution to time-varying overdetermined system of linear equations. *IEEE Trans. Autom. Control* **65**, 874-881 (2020).
31. Zhang, Z. & Zheng, L. A complex varying-parameter convergent-differential neural-network for solving online time-varying complex Sylvester equation. *IEEE Trans. Cybern.* **49**, 3627-3639 (2019).
32. Zhang, Z., Chen, T., Wang, M. & Zheng, L. An exponential-type anti-noise varying-gain network for solving disturbed time-varying inversion systems. *IEEE Trans. Neural Netw. Learn. Syst.* **31**, 3414-3427 (2020).
33. Zhang, Z. & Yan, Z. A varying parameter recurrent neural network for solving nonrepetitive motion problems of redundant robot manipulators. *IEEE Trans. Control Syst. Technol.* **27**, 2680-2687 (2019).
34. Zhang, Z. et al. A varying-parameter convergent-differential neural network for solving joint-angular-drift problems of redundant robot manipulators. *IEEE/ASME Trans. Mechatron.* **23**, 679-689 (2018).
35. Zhang, Z. et al. A varying-parameter adaptive multi-layer neural dynamic method for designing controllers and application to unmanned aerial vehicles. *IEEE Trans. Intell. Transp. Syst.* <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9072299> (2020).
36. Zheng, L., & Zhang, Z. Convergence and robustness analysis of novel adaptive multilayer neural dynamics-based controllers of multirotor UAVs. *IEEE Trans. Cybern.* <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8758355> (2019).

37. Zhang, Z., L. Kong, L., & Zheng, L. Power-type varying-parameter RNN for solving TVQP problems: Design, analysis, and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **30**, 2419-2433 (2019).
38. Zhang, Z. et al. Robustness analysis of a power-type varying-parameter recurrent neural network for solving time-varying QM and QP problems and applications. *IEEE Trans. Syst., Man, Cybern. A, Syst.* **50**, 5106-5118 (2020).
39. Zhang, Z. & Yan, Z. An adaptive fuzzy recurrent neural network for solving the nonrepetitive motion problem of redundant robot manipulators. *IEEE Trans. Fuzzy Syst.* **28**, 684-691 (2020).
40. Zhang, Z., Chen, B., Xu, S., Chen, G. & Xie, J. A novel voting convergent difference neural network for diagnosing breast cancer. *Neurocomputing* **437**, 339-350 (2021).
41. Theodoridis, S. & Koutroumbas, K. *Pattern Recognition (Fourth Edition) Ch. 2* (Academic Press, USA, 2009).
42. Xiao, L. & Zhang, Y. Two new types of Zhang neural networks solving systems of time-varying nonlinear inequalities. *IEEE Trans. Circuits Syst. I-Regul. Pap.* **59**, 2363-2373 (2012).
43. Zhang, Y., Xiao, H., Wang, J., Li, J. & Chen, P. Discrete-time control and simulation of ship course tracking using ZD method and ZFD formula 4NgSFD. In *Proc. IEEE Inf. Technol. Mechatronics Eng. Conf. (ITOEC)* 6-10 (IEEE, 2017).
44. Oza, N. C. & Russhell, S. Experimental comparisons of online and batch versions of bagging and boosting. In *Proc. 7th ACM SIGKDD Intern. Conf. Knowl. Disco. Data Mining* 359-364 (ACM, 2001).
45. Cohen, J. P., Morrison, P. & Dao, L. COVID-19 image data collection. Preprint at <https://arxiv.org/pdf/2003.11597v1.pdf> (2020).
46. Cohen, J. P. et al. COVID-19 image data collection: Prospective predictions are the future. Preprint at <https://arxiv.org/pdf/2006.11988.pdf> (2020).

Author contributions statement

Z.Z. initialized, conceived and supervised the project. Z.Z. and B.C. developed the method. B.C. extracted and preprocessed the data, conducted the experiments, analyzed the results and drafted the manuscript. Z.Z. and J.S. provided constructive suggestions to the manuscript. All the authors provided critical review of the manuscript and approved the final draft for publication.

Funding

This work was supported in part by the National Natural Science Foundation under Grants 61976096, 61603142 and 61633010, Guangdong Basic and Applied Basic Research Foundation under Grant 2020B1515120047, the Guangdong Foundation for Distinguished Young Scholars under Grant 2017A030306009, the Guangdong Special Support Program under Grant 2017TQ04X475, the Science and Technology Program of Guangzhou under Grant 201707010225, the Fundamental Research Funds for Central Universities under Grant x2zdD2182410, the Scientific Research Starting Foundation of South China University of Technology, the National Key Research and Development Program of China under Grant 2017YFB1002505, the National Key Basic Research Program of China (973 Program) under Grant 2015CB351703, Guangdong Key Research and Development Program under Grant 2018B030339001, Guangdong Natural Science Foundation Research Team Program 1414060000024.

Competing interests statement

The authors declare no competing interests.

Figures

Figure 1

Please see the Manuscript PDF file for the complete figure caption

Figure 2

Please see the Manuscript PDF file for the complete figure caption

Figure 3

Please see the Manuscript PDF file for the complete figure caption

Figure 4

Please see the Manuscript PDF file for the complete figure caption

Figure 5

Please see the Manuscript PDF file for the complete figure caption

Figure 6

Please see the Manuscript PDF file for the complete figure caption

Figure 7

Please see the Manuscript PDF file for the complete figure caption

Figure 8

Please see the Manuscript PDF file for the complete figure caption

Figure 9

Please see the Manuscript PDF file for the complete figure caption