Received: 15 October 2021 | Revised: 28 February 2022 | Accepted: 17 April 2022

DOI: 10.1111/coin.12528

SPECIAL ISSUE ARTICLE

Computational Intelligence WILEY

# Real-time COVID-19 detection over chest x-ray images in edge computing

Weijie Xu[1] | Beijing Chen[1,2] | Haoyang Shi[1] | Hao Tian[1] | Xiaolong Xu[1,3]

[1]School of Computer Science, Nanjing University of Information Science and Technology, 210044, Nanjing, China

[2]Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science and Technology, Nanjing, China

[3]State Key Laboratory Novel Software Technology, Nanjing University, Nanjing, China

**Correspondence**
Beijing Chen, Nanjing University of Information Science and Technology, Nanjing 210044, China.
Email: nbutimage@126.com

**Funding information**
National Natural Science Foundation of China, Grant/Award Number: 62072251; Natural Science Foundation of Jiangsu Province, Grant/Award Number: BK20211284

## Abstract

Severe Coronavirus Disease 2019 (COVID-19) has been a global pandemic which provokes massive devastation to the society, economy, and culture since January 2020. The pandemic demonstrates the inefficiency of superannuated manual detection approaches and inspires novel approaches that detect COVID-19 by classifying chest x-ray (CXR) images with deep learning technology. Although a wide range of researches about bran-new COVID-19 detection methods that classify CXR images with centralized convolutional neural network (CNN) models have been proposed, the latency, privacy, and cost of information transmission between the data resources and the centralized data center will make the detection inefficient. Hence, in this article, a COVID-19 detection scheme via CXR images classification with a lightweight CNN model called MobileNet in edge computing is proposed to alleviate the computing pressure of centralized data center and ameliorate detection efficiency. Specifically, the general framework is introduced first to manifest the overall arrangement of the computing and information services ecosystem. Then, an unsupervised model DCGAN is employed to make up for the small scale of data set. Moreover, the implementation of the MobileNet for CXR images classification is presented at great length. The specific distribution strategy of MobileNet models is followed. The extensive evaluations of the experiments demonstrate

© 2022 Wiley Periodicals LLC.

the efficiency and accuracy of the proposed scheme for detecting COVID-19 over CXR images in edge computing.

KEYWORDS

CNN, COVID-19, CXR images, edge computing

## 1 INTRODUCTION

The pandemic of the severe Coronavirus Disease 2019 (COVID-19) has spread globally since January 2020.[1] According to the data as received by World Health Organization (WHO) from national authorities by 10:00 CEST, July 25, 2020, 15,581,009 identified cases of COVID-19 have been reported so far all over the world, including 635,173 deaths and impacting all walks of life in society.[2] Currently, the reverse-transcription polymerase chain reaction (RT-PCR) is considered as an appropriate approach for detecting the COVID-19.[3] However, the detection efficiency and accuracy are limited by the deficiency of manpower and material resources when utilizing RT-PCR for screening suspected subjects. Furthermore, RT-PCR testing suffers from the approximately 30% rate of making false-negative diagnoses and the long interval between being tested and obtaining results.[4] Powered by the development of radiological imaging techniques, the chest x-ray (CXR) which demonstrates effectiveness and accuracy in diagnosing, assessing, and evaluating plays a crucial complement to RT-PCR testing. More specifically, compared to RT-PCR results, 0.97 of sensitivity, 0.25 of specificity, and 0.68 of accuracy for the detection of COVID-19 are accomplished by CXR diagnoses.[5]

Although the utilization of CXR for COVID-19 detection reduces the probability of false-negative diagnoses, diminishes the interval of waiting for results, and improve the efficiency of detection, the manual delineation and discrimination of the pandemic is a monotonous and prolonged assignment. Additionally, pandemic annotation by radiologists is a highly subjective task, often influenced by individual bias and clinical experiences. To make up for the defect of manual diagnosis, numerous approaches which detect the COVID-19 by classifying CXR images with deep learning technologies have been proposed recently. For instance, a confidence-aware anomaly detection model which is composed of multiple modules was proposed to automatically analyze the abundant plentiful CXR images.[6] A tailored deep convolutional neural network (CNN) named as COVID-Net was designed for the detection of COVID-19 cases from CXR images.[7] A list of paper that conducted researches on CXR images to solve problems brought by COVID-19 were introduced in a survey.[8]

Whereas, the existing researches are mainly conducted in a centralized way, which is subjected to the limited computing resources, frequent data exchanges, risk of information leakage, and high latency of information transmission between data resources and centralized data center. In the absence of medical supplies, it is almost impossible to be provided with sufficient computing resources for COVID-19 detection. For example, in a temporary hospital (e.g., mobile cabin hospital), suspected or observed patients are hardly treated with adequate medicine, to say nothing of building a centralized data center for storing CXR images. As a result, it is unfeasible to train a comprehensive, exhaustive, and complete CXR image classification model in real-time. According to the report made by WHO, the investigators conducted researches on the genetic sequence of the virus and concluded that the COVID-19 may evolve over time. At the present stage, radiography findings mainly include bilateral patchy shadows or ground-glass opacity of the lungs on chest computed tomography scan. If the COVID-19 evolves and changes its appearance on CXR, the inference model trained with the previous data set will no longer be applicable to the evolved COVID-19.

To tackle the issues mentioned above, we propose a COVID-19 detection manner that classifies CXR images with CNN in edge computing. More concretely, the main goal of presenting this manner is to accelerate the COVID-19 detection in computing-limited medical environments and keep the model valid even if the COVID-19 evolves. To this end, a CNN model is implemented for COVID-19 detection by classifying CXR images, so as to keep a balance between the model dimension and the detection precision. Additionally, the overall framework takes the cost, privacy, and latency of information transmission into consideration. Furthermore, to ensure the accuracy of models in edge devices, an optimization scheme is designed at great length. In essence, the dominating contributions made in this article can be summarized as follows:

- A lightweight CNN model called MobileNet is implemented for COVID-19 detection by classifying CXR images, so as to keep a balance between the model dimension and the detection precision when the model works in edge devices.
- The transmission latency and privacy of information transmission are taken into consideration when designing the framework of models in the edge devices.
- Generative adversarial networks (GAN), an unsupervised learning model, is utilized to raise the accuracy of lightweight models in edge devices.
- The experimental results demonstrate that the optimized MobileNetV2 achieves an average precision of 82% with the lightest model size of 13.6 MB.

The remaining part of this article is organized as follows. First, the retrospect of related work is conducted in Section 2. Then, the overall framework of the system model and the problem definition are introduced in Section 3. The presentation of proposed schemes, including the implementation of a lightweight CNN model called MobileNet for COVID-19 detection, the framework of models in edge devices, and the unsupervised learning model for optimization are followed in Section 4. We evaluate the performance of the experiments in Section 5 and conclude this article in Section 6.

## 2 | RELATED WORK

Three types of related work that pertain to our work are discussed in this section, including classification of CXR images with CNN, the model arrangement in edge devices, and unsupervised learning.

### 2.1 | CNN for CXR image classification

Driven by the marvelous development of hash rate, deep learning plays a pivotal role in medical image analysis such as computed tomography (CT) utilized for the diagnosis of multiple diseases in different organs and x-ray employed for detecting lung diseases commonly.[9] It is universally utilized in finding or learning informative features that well describe the regularities or patterns inherent in data, so as to provide crucial information for diagnoses.[10] Plentiful investigations are conducted to consummate the classification algorithms. For illustrative purposes, Marlon et al.[11] proposed a method utilizing CNN named TX-CNN to classify unbalanced, less-category x-ray images with considerable accuracy and efficiency. Joseph et al.[12] presented an implementation of CNN called XNet for medical x-ray image classification suitable for small datasets. However, affected by the image quality of CXR, a few CNN models fail to meet the accuracy requirements of COVID-19 detection. To conquer the serious issue, Kabid et al. introduced a deep neural network (DNN) based faster regions with convolutional neural networks (Faster R-CNN) framework to detect COVID-19 from CXR images. Besides, Zabirul et al.[13] combined the CNN with long short-term memory (LSTM) to detect COVID-19 automatically from CXR images. Specifically, CNN was utilized for deep feature extraction by classifying CXR images while the LSTM was applied for detection using these features. In addition to the problem of accuracy, COVID-19 detection suffers from insufficient data in the dataset at the very beginning of the pandemic. Gaurav et al.[14] utilized transfer learning (TL) as a supplement to CNN for COVID-19 detection from CXR images, so as to overcome the lack of COVID-19 data. Meanwhile, Tawsifur et al. integrated TL with several pre-trained CNN (e.g., AlexNet, ResNet18, DenseNet201, and SqueezeNet) for higher accuracy.

## 2.2 | Edge computing

The emergency treatment of COVID-19 is critically constrained by limited human and material resources. Although a certain extend of work done by doctors is taken by medical equipment, the equipment operating efficiency and response rate suffers from insufficient communication and computing resources.[15] Fortunately, edge computing infrastructure will improve network efficiency while reducing the amount of mobile data. Each local network component processes part of the information it collects. Therefore, edge computing reduces the reliance on remote centralized servers or distributed local servers, so that temporary hospitals and clinics gain more agile and responsive networks. Chaitra et al.[16] constructed an open-source edge computing raspberry pi-based clinical screening system named AutoTriage. The AutoTriage real-time detects the presence and body parts in the view of the surveillance camera, and classifies out the forehead and lip regions, so as to determine whether the individuals' body temperature is abnormal. Mohammad et al.[17] designed a reorganizing biosurveillance framework for the detection and localization of biological threats with fog and mobile edge computing support. The framework not only alerts individuals if they contact someone who is positive for COVID-19 but also detects biological threats with edge computing before being clinically recognized. The tasks mentioned above tend to carry various private information such as patients' personal information, and training tasks. To protect the privacy of information utilized in edge computing, Whaiduzzaman et al.[18] presented a privacy-preserving mobile and fog computing framework to trace and prevent COVID-19 community transmission.

## 2.3 | Unsupervised learning

The spread and outbreak of new coronary pneumonia are extremely fast. As a result, it is challenging to accumulate sufficient CXR images to train a model with a high confidence level. Unsupervised learning is a type of machine learning technique utilized to find patterns in data. It is essentially a statistical method in which some potential structures could be found in unlabeled data. GAN is one of the classic unsupervised learning methods which intuitively understands the features learned by the encoder to reconstruction images. Bao et al.[19] presented a general learning framework that combines a variational auto-encoder with a GAN for synthesizing images in fine-grained categories. The framework named CVAE-GAN is applied to several tasks including image inpainting, super-resolution, and data augmentation for training better models. Besides, Changhee et al.[20] proposed an approach that generates synthetic multi-sequence brain magnetic resonance images using GANs. This contribution reminds researchers of expanding the image dataset by generating images through GAN. To optimize the effect of the GAN model that generates the image, Zhai et al.[21] studied lifelong learning for generative models, extending a trained network to new conditional generation tasks without forgetting previous tasks.

# 3 | SYSTEM MODEL AND PROBLEM DEFINITION

In this section, the system model of COVID-19 detection from CXR Images with CNN in edge computing is designed in Section 3.1, while the problem is defined in Section 3.2. The notations in this article are given in Table 1 with their definitions and functions.

## 3.1 | Framework of COVID-19 detection from CXR images with MobileNet in edge computing

The system model is proposed as an instance of COVID-19 detection from CXR Images with CNN in edge computing, which is a new and effective computing paradigm for distributed data sharing

TABLE 1 Notations and definitions of the framework components

| Notation | Definition |
|---|---|
| $H$ | The set of hospitals, $H\{h_1, h_2, \dots, h_n\}$. |
| $C$ | Collection of DS, DP, and CD. |
| $D$ | The set of distances, $D = \{dist_{C,h1}, dist_{C,h2}, \dots, dist_{C,hn}\}$. |
| $P$ | The set of privacy entropy, $P = \{p_1, p_2, \dots, p_n\}$. |
| $M$ | The set of models, $M = \{m_1, m_2, \dots, m_n\}$. |
| $X$ | A parameter ranges from 0 to 1. |
| $W$ | The set of weight, $W = \{W_1, W_2\}$. |
| $S$ | Speed of information transmitted in medium. |
| $N$ | The amount of data to be processed. |
| $R$ | The request speed of data. |
| $T_1$ | The informational transmission time. |
| $T_2$ | The transmission delay. |
| $I$ | Importance of the cost time. |
| $T_t$ | Total time of the transmission latency. |

and processing in an edge computing scenario. As is shown in Figure 2, the system consists of a data processor (DP), a centralized database (CD), a data switch (DS), multiple hospitals including a temporary hospital (TH), local clinic (LC), and regular hospital (RH). Patients who are suspected or observed will gather at the adjacent CXR inspection site to get real-time CXR images. Here, the DP, CD, and DS are located in the cloud and the hospitals are spatially distributed in different regions. The functions of these framework components are shown in Table 1.

More specifically, an example is given to explain the operating process of the system model.[22] In Figure 1, patients who are suspected or observed gather at the adjacent CXR inspection site such as TH, LC, and RH to have CXR checks. The CXR images are uploaded by the DS to the centralized data center. Then the DS transmits the collected data to the CD for storage. Afterwards, the DP preprocesses the data stored in CD by cleaning the raw data. While the clean data is updated to the CD and performs as the original dataset, the DP trains a GAN model composed of a generative model and a discriminative model with the original dataset. The size of the dataset is expanded by the CXR images generated from the GAN model. The GAN implemented in this article is not the traditional GAN model which utilizes functions that fit corresponding generated and discriminant, but employs CNNs to perform the role of fitter. With such a GAN called deep convolutional generative adversarial networks (DCGANs), the output CXR images owns higher confidence than those generated by traditional GAN. Next, the newly expanded dataset is utilized to train a lightweight CNN model called MobileNet for the classification of CXR images in edge devices.[23] The MobileNet reduces parameter size while maintaining model accuracy and guaranteeing the low latency of the model. Finally, due to the centralized data center takes the model size, edge computing capabilities, privacy entropy into consideration to offload the models through the DS in an efficient way. The lightweight MobileNet models deployed in edge devices are able to perform reasoning tasks with limited computing resources. As a result, the detection tasks are accomplished without transmitting CXR images, so as to diminish the total time of the detection.

## 3.2 | Problem definition

In the COVID-19 detection system, the predominant goal is to scale down the model size and reduce the cost of model offloading while ensuring the accuracy of the model.[24] Hospitals in different regions or different levels of hospitals in the same area have different computing resources. Besides, the distances between edge devices in hospitals and centralized data center may conduct a certain impact on the data transmission efficiency and cause potential user or model privacy leakage issues. The hospitals, denoted by set $H = \{h_1, h_2, \dots, h_n\}$ where $n$ stands for the number of the hospitals, are divided into several categories including THs, LCs, and RHs. Different amounts of computing resources are occupied by various classifications of hospitals. The centralized data center, denoted by set $C$, owns a fixed amount of resources which is denoted as $S$. The distance between the $H$ and $C$, denoted by set $D = \{dist_{C,h1}, dist_{C,h2}, \dots, dist_{C,hn}\}$, is related to energy consumption, time-consuming, and safety of information transmission. $S$ is the transmission rate in the corresponding transmission medium. $N$ is the total number of bits of the transmission task. $R$ is the data processing rate. The resource $X$ of privacy is expressed as

$$\begin{pmatrix} X \\ P(X) \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ p(x_1) & p(x_2) & \dots & p(x_n) \end{pmatrix}, \tag{1}$$

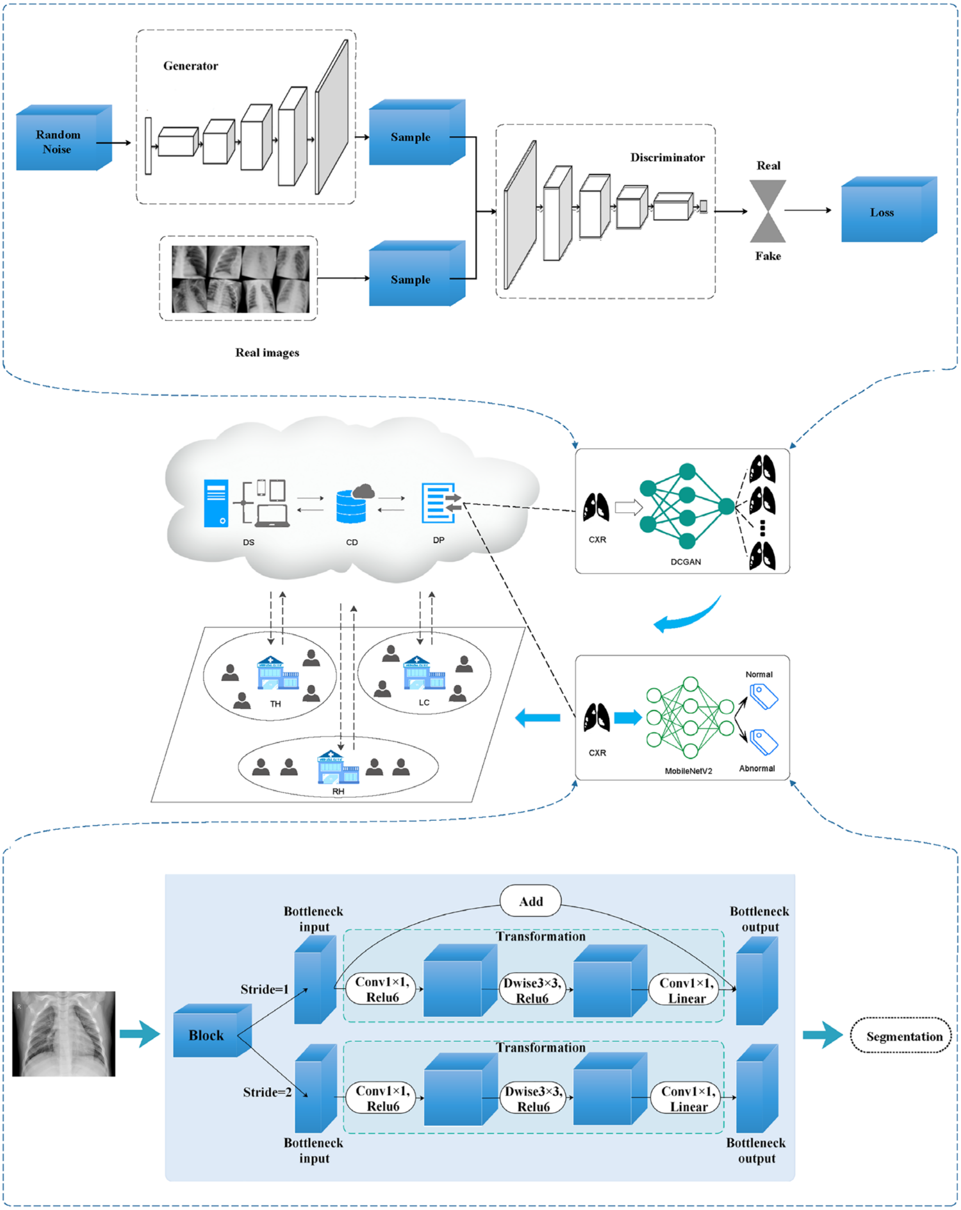


FIGURE 1 The framework of COVID-19 detection from CXR images with CNN in edge computing.

while similarly the resource $Y$ of privacy is shown as

$$\begin{pmatrix} Y \\ P(Y) \end{pmatrix} = \begin{pmatrix} y_1 & y_2 & \dots & y_n \\ p\,(y_1) & p\,(y_2) & \dots & p\,(y_n) \end{pmatrix}. \tag{2}$$

Meanwhile, the privacy entropy is calculated by

$$h(X) = -\sum_{i=1}^{n} p(x)log_2(p(x)). \tag{3}$$

Two parts (i.e., transmission time between the centralized data center and edge devices, transmission delay) make up the main transmission latency.[25] The transmission time is closely related to the distance between the centralized data center and edge devices. The distance is calculated by

$$dist_{C,hn} = \sqrt{(x_c - x_{hn})^2 + (y_c - y_{hn})^2}. \tag{4}$$

Then the transmission time $T_1$ is calculated by

$$T_1 = \frac{dist_{C,hn}}{S}, \tag{5}$$

while the transmission delay $T_2$ is calculated by

$$T_2 = \frac{N}{R}. \tag{6}$$

In summary, the total latency of transmission $T$ is calculated by

$$T_t = \sum_{n=1}^{2} I_n T_n. \tag{7}$$

The privacy entropy of each model, denoted by set $P = \{p_1, p_2, \dots, p_n\}$, is a quantitative judging criterion of privacy, is leveraged to measure the data transmission security. The offloading problem of the model set $M\{m_1, m_2, \dots, m_n\}$ is defined as a multi-objective optimization (MOO) problem which takes the privacy entropy and transmission latency into consideration.

## 4 | PRELIMINARY KNOWLEDGE

In this article, two kinds of crucial preliminary knowledge are necessitated.[26] An unsupervised learning which combines CNNs with the GANs called DCGAN is utilized for generating more CXR images utterly different from the original ones.[27] Meanwhile, a lightweight CNN model called MobileNet is employed with its second version to classify the CXR images, so as to detect COVID-19 from them.

### 4.1 | DCGAN

GAN, a classic deep learning method, is one of the most promising approaches for unsupervised learning on complex distributions.[28] Standard GAN generates satisfying output through mutual game learning between at least two types of modules in the overall framework: generative model and discriminative model. The generative model is responsible for capturing data distribution while the discriminative model is in charge of estimating the probability that the sample comes from the training data. In other words, the training procedure of the generative model maximizes the error probability of discriminative. Beyond that, the training tasks can be conducted with back propagation if both of the generative model and discriminative model are defined by multi-layer perceptrons.

DCGAN is a special GAN which replaces the traditional generative multilayer perceptron and discriminative multilayer perceptron with two convolutional architecture. Additionally, the generative model of DCGAN utilizes a special structure called transposed convolution creatively.[29] Furthermore, the Relu is widely employed in the activation function of the generator except the last layer. The last layer of generative model is activated by tanh instead while the discriminative model is totally activated by Leakly-Relu function.[30]

In DCGAN, $p_{data}$ refers to a generating variable about the data $x$. $z$ is random noise added to the generator $G(z)$ with the real world images. The generative model has relativity to the argument $\theta_g$ while the discriminative model $D(x)$ is related to the parameter $\theta_d$. Moreover, the discriminator takes in either real world images or synthetic ones and then outputs confidence that $x$ came from the real world images data set.[31] The main purpose of generator is generating images as real as possible, so as to deceive the discriminator. Likewise, the discriminator aims to separate the images generated by generator from the real ones. In a word, the $G(z)$ and $D(x)$ formulate a dynamic gaming process with value function $V(G, D)$:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]. \tag{8}$$

During the process, the $D$ trends to aggrandize the value of $V$ while the $G$ attempt the opposite. So, first, the $D$ is trained with the $G$ locked, just like

$$\max_D (\mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z))]). \tag{9}$$

Then the $D$ is restrained for training the $G$ by

$$\max_G \left(\mathbb{E}_{z \sim p_z(z)}[\log(D(G(z))]\right) . \tag{10}$$

When the $p_{data}(x)$ is equal to the $p_g(x)$, the optimal $G$ will be received as well.

## 4.2 | MobileNet

MobileNet is a lightweight CNN model on behalf of the unexceptionable development of intelligent edge.[32] The marvelously small size of MobileNet makes it possible to be deployed in the edge devices, so as to promote the process of edge intelligence. Specially, in the area of studying image classification, the MobileNet plays a crucial role in the application of edge vision.

The MobileNet has two optimized versions. The overall structure of MobileNetV1 is similar to the straight-through structure of general neural networks.[33] However, the MobileNetV1 is different from the previous neural networks mainly in two aspects. First, the depthwise separable convolution is used to improve the calculation speed of the network. The depthwise separate convolution includes depthwise convolution and pointwise convolution. Besides, MobileNetV1 adopts the width multiplier, which is simply to introduce a new hyper-parameter to adjust the number of channels of the convolution output, so as to conveniently balance the calculation speed and accuracy of the network. For further majorization of the model effectiveness, the

MobileNetV2 proposed several optimize points on the basis of inheriting the depthwise separable convolution.[34] Specifically, the optimization is mainly operated in three aspects. First, for the low-latitude space output by Relu, linear convolution is utilized to extract features in order to reduce feature loss.[35] Then, for the layer with Relu, in order to prevent too much feature loss, first increase the number of feature channels.[36] Finally, referring to the ResNet structure, the shortcuts connection method is also added to slow down the gradient dispersion caused by BP. The basic structure of MobileNetV2 and ResNet is very similar. However, ResNet first reduces the dimensionality to improve the features. Then it increases the dimensionality while In MobileNetV2, the dimensionality is first upgraded, so that features are improved, and then dimensionality is reduced.

The calculation ratio of standard convolution $S_c$ is represented as

$$S_c = D_K \times D_K \times M \times N \times D_F \times D_F, \tag{11}$$

while the calculation ratio of depthwise convolution $D_c$ is calculated by

$$D_c = D_K \times D_K \times M \times D_F \times D_F, \tag{12}$$

Meanwhile, the calculation ratio of pointwise convolution $P_c$ is formulated as

$$P_c = M \times N \times D_F \times D_F, \tag{13}$$

So, the overall calculation ratio of depthwise separable convolution $A_c$ is

$$A_c = D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F, \tag{14}$$

As a result, the comparison result of standard convolution and depthwise separable convolution $C$ is

$$C = \frac{1}{N} + \frac{1}{D_K^2}. \tag{15}$$

Among these parameters, $D_F$ means the width and height of the feature. The $D_K$ refers to the side length of the filters. $N$ means the number of filters while the $M$ refers to the standard convolution filters depth.

Benefit from the special structure, MobileNetV2 reduces parameters size while maintaining model accuracy.

## 5 | PROPOSED SCHEME

In this section, the schemes utilized in this article are introduced at great length. The unsupervised optimization model DCGAN is presented in Section 5.1, while the MobileNet for COVID-19 detection by classifying CXR images is instituted in Section 5.2. The MOO of model offloading in edge computing is introduced in Section 5.3. The overall schemes proposed in this article is shown in Figure 2.

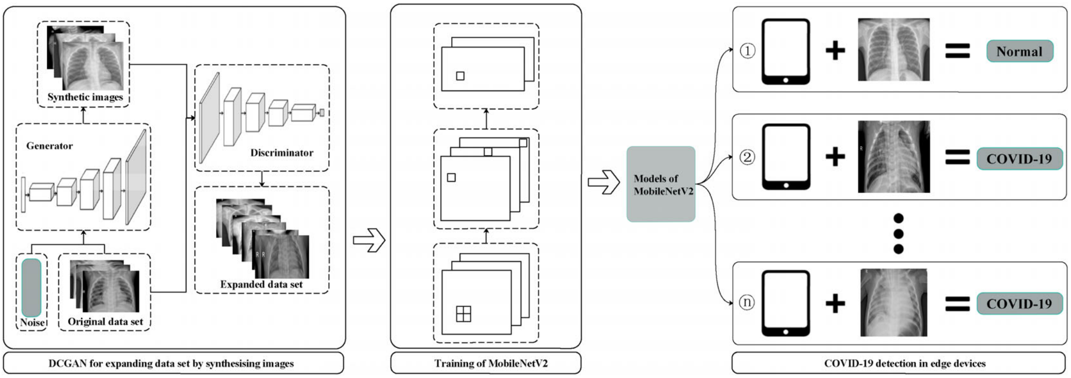


FIGURE 2 The overall scheme utilized.

## 5.1 | Unsupervised optimization model-DCGAN

The pandemic breaks out so fast that there is not enough dataset to train a high-accuracy discriminant model. GAN is an unsupervised learning model that can be utilized to generate more images with the original dataset.

For further improving the model effect, the DCGAN implemented in this article is accommodated according to the dataset. The DCGAN architecture with no tanh is utilized in the generator, while the ELU is treated as the discriminator. The ELU is calculated as

$$f(z) = \begin{cases} z & z > 0, \\ \alpha(exp(z) - 1) & z \leq 0. \end{cases} \tag{16}$$

In the process of implementing the DCGAN in this article all filters are consistently set to $4 \times 4$. Besides, channel is set as a half size of the filters as well for DCGAN training. A batch size of 32 and optimizer with a maximum learning rate of $1.0 \times 10^{-4}$ are implemented. Among these components of DCGAN, the forward propagation of deconv is extraordinarily similar to the backward propagation of conv. If the generator is observed from back to front, it is a typical CNN, using conv convolution, the feature map gradually decreases while the number of channels gradually increases. More specifically, the input of generator is random noise which would be reshaped to $4 \times 4 \times 1024$. After a $5 \times 5$ deconv, the size will become $8 \times 8 \times 512$. And so on, the final output is $64 \times 64 \times 3$. Similarly, the input of discriminator is $64 \times 64 \times 3$, and then processed to $32 \times 32 \times 64$ by a $5 \times 5$ conv.

## 5.2 | MobileNet for COVID-19 detection by classifying CXR images in edge devices

The manual inspection of COVID-19 is inefficient and unfaithful while the currently available automatic detection approaches own marvelous accuracy but cost much computing resources and information transmission materials. This scenario requires low latency, fast response speed, and high accuracy with a model that is minuscule enough to be applied in mobile or embedded devices for edge computing. To achieve a balance among these elements, a lightweight model named MobileNet is utilized for CXR image classification in edge devices.[32] In the block whose stride is 1, the flow is divided to 4 steps, while in the block whose stride is 2, the flow consists of 3 parts.

More specifically, in the case of stride equal to 1, the input CXR image is treated as the bottleneck input. Then the bottleneck input will be transformed to the bottleneck output which means the consequence of classification. During the transformation, the input first goes through a Conv layer sized 1 × 1 occupied with a Relu6. The treated data enter the Dwise layer sized 3 × 3 with Relu6. Finally, the data turn into the Conv layer sized 1 × 1 with linear. When the stride is 1, the output may also be got by adding the input to a shortcut.

## 5.3 | MOO of model offloading for edge computing

The information transmitted between the centralized data center and edge computing devices always carries a wide range of data which may bring about grave loss if it is disbosomed. Numerous indicators have been utilized to judge the security of information transmission, including difference degree, variance, privacy entropy, degree of anonymization, and risk of data leakage. Meanwhile, the latency of transmission deserves being taken into consideration. In a nutshell, the latency and privacy entropy of models offloaded to edge devices are considered as a MOO problem that aims at the best offloading strategy.[37]

The privacy entropy, a quantitative approach, is leveraged to measure the security of data transmission.[38] It is utilized to evaluate the unpredictability of a data value which means the difficulty of predicting the original value of the data after privacy protection processing. Because entropy represents the amount of information of the data, the entropy of the data after privacy protection processing should be higher than the entropy before processing.[39] To protect privacy during the data transmission, privacy entropy must be taken into consideration.

The premier grail of the model offloading MOO problem is retrieving the Pareto optimal (PO). PO refers to an ideal state of resource allocation.[40] Given an inherent group of individuals and assignable resources, if the change from one distribution state to another state makes at least one person better without making anyone worse off, this is Pareto improved.[41] The optimal state of Pareto is that there can be no more Pareto improvements. In other words, the core ideas can be condensed into 2 points. The one is finding the solution set as close to the Pareto optimal front as possible, while the other is maximizing the diversity of solutions that would be found.

An algorithm named Predominant Naïve Pareto optimal (PNPO) which combines multiple weights with various influence factors is utilized for seeking out the PO. The PNPO algorithm describes the procedures that consider both of privacy entropy and transmission latency. The core formulation of PNPO is calculated as

$$\min F(x) = \sum_{n=1}^{2} W_n f_n(x). \tag{17}$$

Figure 3 illustrates the overall procedures of the COVID-19 detection framework. The procedures start from the initialization of the $G(z)$ and $D(x)$. Then the generator and discriminator take a game for generating CXR images with high confidence that are completely different from the original images. Afterward, the expanded data set with generated CXR images is employed for training the MobileNetV2. The trained MobileNet models are offloaded to edge devices with the PNPO algorithm.

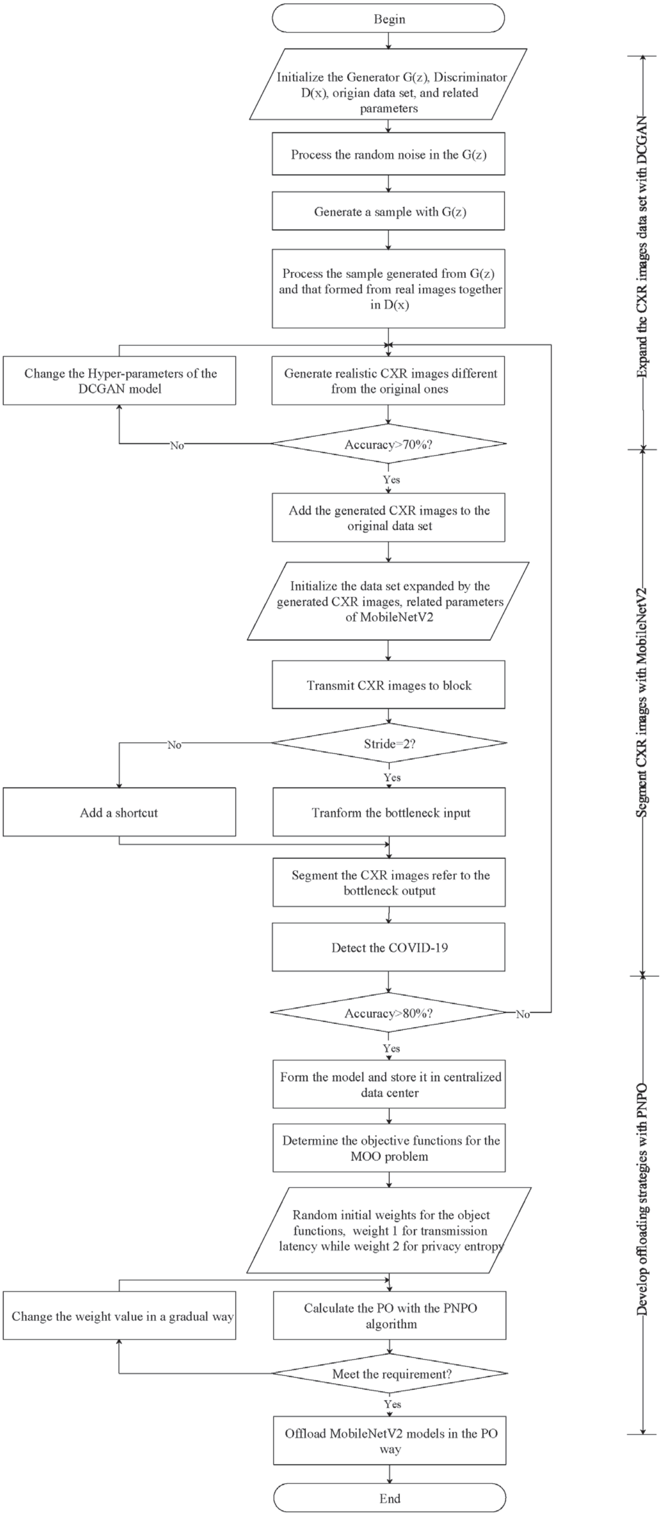

Begin
Initialize the Generator G(z), Discriminator D(x), origian data set, and related parameters
Process the random noise in the G(z)
Generate a sample with G(z)
Process the sample generated from G(z) and that formed from real images together in D(x)
Change the Hyper-parameters of the DCGAN model
Generate realistic CXR images different from the original ones
No
Accuracy>70%?
Yes
Add the generated CXR images to the original data set
Initialize the data set expanded by the generated CXR images, related parameters of MobileNetV2
Transmit CXR images to block
No
Stride=2?
Yes
Add a shortcut
Tranform the bottleneck input
Segment the CXR images refer to the bottleneck output
Detect the COVID-19
Accuracy>80%?
No
Yes
Form the model and store it in centralized data center
Determine the objective functions for the MOO problem
Random initial weights for the object functions, weight 1 for transmission latency while weight 2 for privacy entropy
Change the weight value in a gradual way
Calculate the PO with the PNPO algorithm
Meet the requirement?
Yes
Offload MobileNetV2 models in the PO way
End
Expand the CXR images data set with DCGAN
Segment CXR images with MobileNetV2
Develop offloading strategies with PNPO

FIGURE 3 The overall procedures of the COVID-19 detection framework

# 6 | PERFORMANCE EVALUATION

## 6.1 | Experiment setup

In the experiment of this article, a Dell workstation occupied with a GPU of Nvidia RTX 2080ti and an 8 core CPU is utilized to simulate the status of the centralized data center while virtual machines (VMs) with TPU are employed as the edge computing environment.[42] The original dataset of CXR images fails to realize a lightweight MobileNetV2 model with suitable accuracy, so we extend the dataset with the DCGAN by generating images completely different from the original CXR images. Additionally, three typical models that utilized for image classification are compared in this article, including MobileNetV2, ResNet18, and VGG19. They are compared to prove the applicability of MobileNetV2 in the proposed scene.

The important hyper-parameter settings of the experiment are shown in Table 2. The upper learning rate is set as 0.0001 to ensure that the neural network converges to the global minimum, so as to avoid undesirable consequences on the loss function. The epoch is chosen as 20 to keep a balance between the training time cost and the model effect. The weight decay is 1e-4 which is able to prevent overfitting. The Adam is employed as the optimizer because it is easy to implement, owns high computational efficiency, and requires low memory. The batch-size has a great influence on the optimization and speed of the model. So we test the batch-size of 16, 32, 64, and 128 to find the best balance between memory efficiency and memory capacity. When working in the environment set up in this article, the 32 batch-size performs best in the case of considering the balance between the model accuracy and training cost.

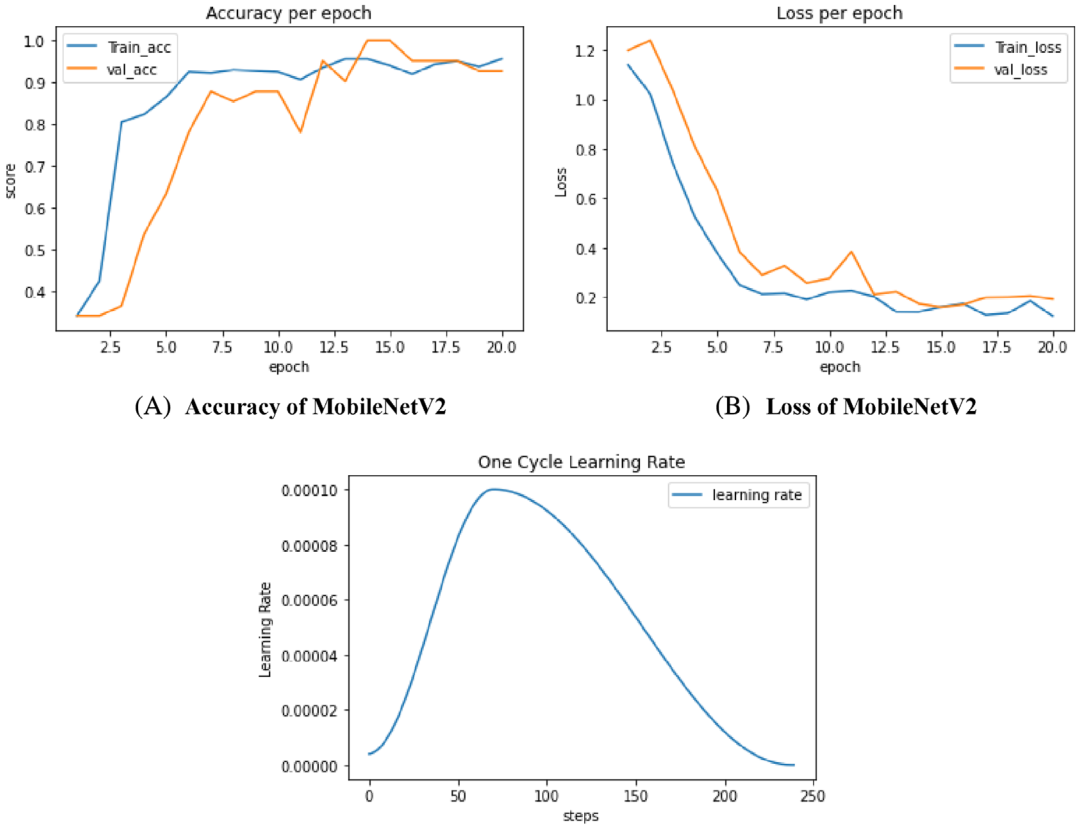
## 6.2 | Experimental evaluation

In this article, three typical models are compared, including MobileNetV2, ResNet18, and VGG19, to find out the most suitable one. The accuracy, loss, and learning rate of training tasks are compared in Figures 4–6, respectively. All of the three models receive a high rate of accuracy, pavely low loss, and stable learning rate. Specifically, the MobileNetV2 keeps a best balance of model accuracy and model size.

Besides, the parameters serving as criteria of the three models are listed in Table 3. The support means the number of CXR images utilized for COVID-19 detection in the experimental evaluation. The overall support of number 22 is divided into three parts which is 7, 7, and 8, respectively, so that the evaluation is conducted more efficient. Each support group is evaluated to get the precision, recall, and f1-score. What's more, the average accuracy, macro average, and weighted average

TABLE 2 Hyper-parameter settings of the experiment

| Hyper-parameter | Value |
|---|---|
| Upper learning rate | 0.0001 |
| Epoch | 20 |
| Weight decay | 1e-4 |
| Optimizer | Adam |
| Scheduler | One cycle |
| Batch size | 32 |



(A) **Accuracy of MobileNetV2**

(B) **Loss of MobileNetV2**

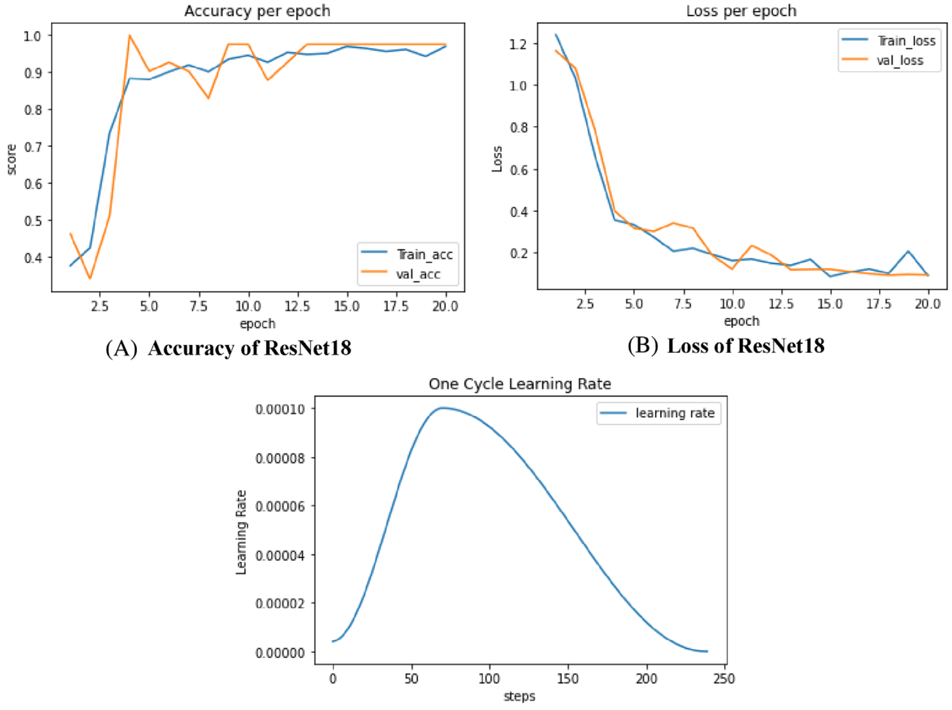
(C) **Learning rate of MobileNetV2**

FIGURE 4 (A) Accuracy, (B) loss, and (C) learning rate of MobileNetV2

are evaluated as well. Among the three models utilized for comparison, the MobileNetV2 fails to receive the highest average accuracy but it receives the uppermost precision 0.92 and F1-score 0.93. Moreover, the model size of MobileNetV2 is 13.6M while that of ResNet18 and VGG19 are 44.7M and 548M. The MobileNetV2 scale is only 30% of ResNet18, even only 2% of VGG19. The model mass of MobileNetV2 is the tiniest among three models, so the transmission cost and reasoning time of MobileNetV2 are low. The recall and F1-score of MobileNetV2 also perform better that of the other two model.

A group of six CXR images are utilized for testing the detection accuracy. The testing result of MobileNetV2, ResNet18, VGG19 are shown in Figures 7–9, respectively. The result demonstrates that all of the three models are able to detect the COVID-19 by classifying CXR images in a functionally high accuracy. The CXR images chosen to be detected are diagnosed precisely to show if the patients are infectious.

When detecting COVID-19 with these models, the time cost is hardly perceptive for human. As a result, a set of CXR images which owns an amount of at least 100 is utilized for evaluating the detecting efficiency. In the process of detecting 100 CXR images, the cost of time can be recorded within the latency perceived by individuals. Testing sets are chosen as five types (i.e., 100, 200, 300, 400, and 500). The time consumption is concluded in Figure 10. Among the three compared models, the MobileNetV2 takes the least time to detect the same amount of CXR images at once,

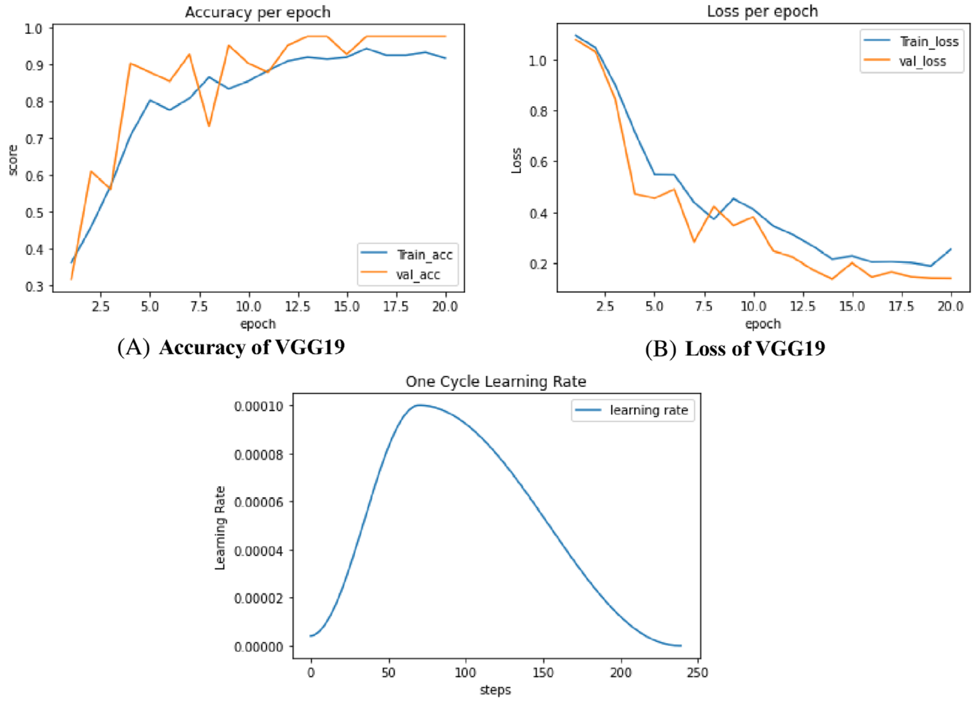


(A) **Accuracy of ResNet18**

(B) **Loss of ResNet18**

(C) **Learning rate of ResNet18**

FIGURE 5 (A) Accuracy, (B) loss, and (C) learning rate of ResNet18



(A) **Accuracy of VGG19**

(B) **Loss of VGG19**

(C) **Learning rate of VGG19**

FIGURE 6 (A) Accuracy, (B) loss, and (C) learning rate of VGG19

**TABLE 3** Parameters serving as criteria of comparing the three models

| Model name | Options | Precision | Recall | F1-score | Support | Model size |
|---|---|---|---|---|---|---|
| MobileNetV2 | 0 | 0.83 | 0.71 | 0.77 | 7 | 13.6 MB |
| | 1 | 0.87 | 0.86 | 0.75 | 7 | |
| | 2 | 0.92 | 0.82 | 0.93 | 8 | |
| | Accuracy | | | 0.82 | 22 | |
| | Macro avg | 0.83 | 0.82 | 0.82 | 22 | |
| | Weighted avg | 0.84 | 0.82 | 0.82 | 22 | |
| ResNet18 | 0 | 0.88 | 0.82 | 0.88 | 7 | 44.7 MB |
| | 1 | 0.83 | 0.86 | 0.82 | 7 | |
| | 2 | 0.87 | 0.97 | 0.92 | 8 | |
| | Accuracy | | | 0.87 | 22 | |
| | Macro avg | 0.86 | 0.87 | 0.87 | 22 | |
| | Weighted avg | 0.86 | 0.87 | 0.87 | 22 | |
| VGG19 | 0 | 0.88 | 0.83 | 0.79 | 7 | 548 MB |
| | 1 | 0.83 | 0.82 | 0.80 | 7 | |
| | 2 | 0.85 | 0.86 | 0.91 | 8 | |
| | Accuracy | | | 0.83 | 22 | |
| | Macro avg | 0.84 | 0.86 | 0.86 | 22 | |
| | Weighted avg | 0.84 | 0.86 | 0.86 | 22 | |

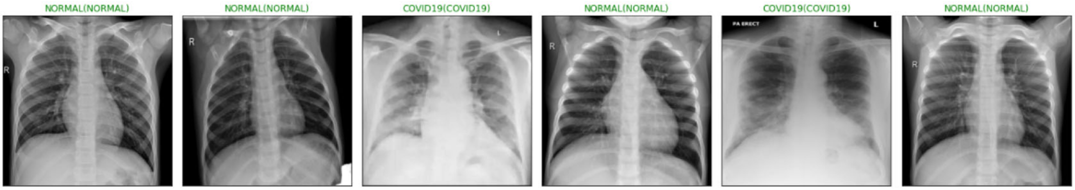


**FIGURE 7** Detection test of MobileNetV2

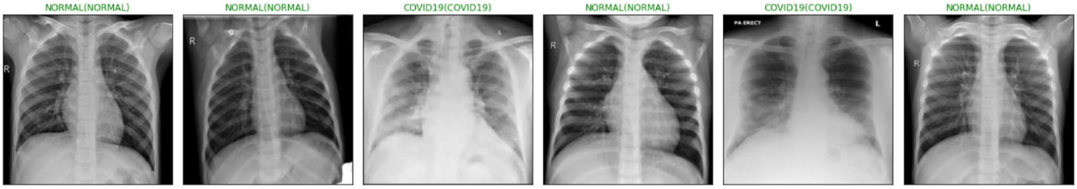


**FIGURE 8** Detection test of ResNet18

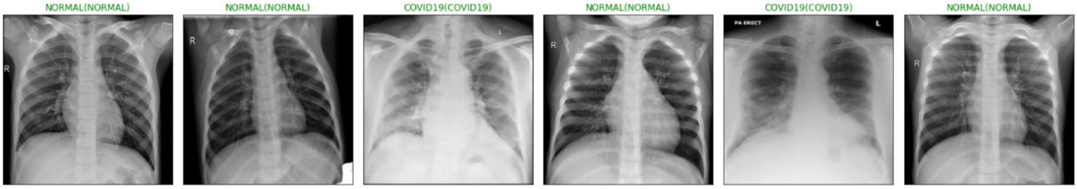


FIGURE 9 Detection test of VGG19

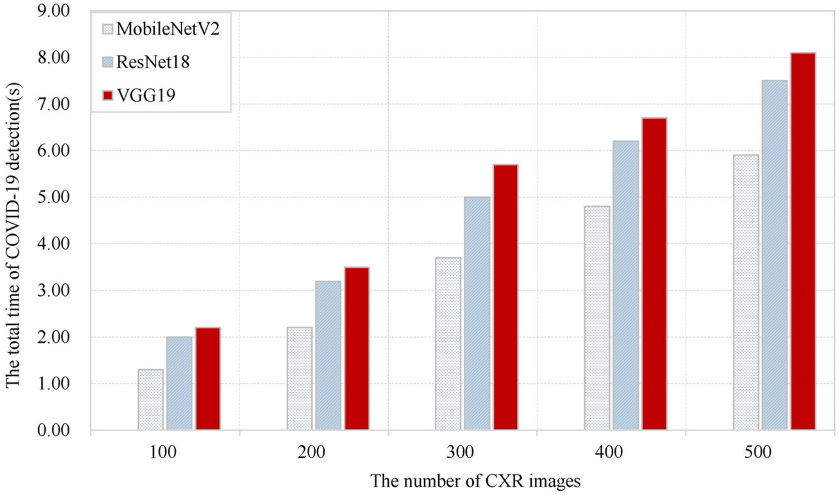


FIGURE 10 Total time consumption of detecting COVID-19 from different numbers of CXR images.

because of the lightest model magnitude. Besides, it can be concluded from the figure that the detection time grows as the model size increases.

After all, the ResNet18 solves the degradation problem of DNN and the problem of deep network gradient disappearance with its extremely special residual structure, thus obtaining a highly accurate model. However, a particularly large number of network layers in ResNet18 results in a consequence that deep networks tend to take quite a long time to train. Therefore, the cost of applying it to actual scenarios is very high. Meanwhile, the structure of VGG19 is very simple, the entire network employs the same size of the 3×3 convolution kernel layer and 2×2 maximum pooling layer. Besides, the performance will keep improving if the number of layers continues to raise. But VGG19 consumes more computing resources and utilizes more parameters which results in more memory usage because of it multiple fully connected layers that carry too many parameters. MobileNetV2 is characterized by configurable parameters, low latency, reduced memory consumption, and efficient operation. Though the accuracy of MobileNetV2 may not be higher than the other two models, the MobileNetV2 is still the most suitable one for utilization in edge environments.

The overall model transmission latency is shown in Figure 11 when transmitted to a different number of edge devices (e.g., 50, 100, 150, 200, 250) while the privacy entropy is illustrated in Figure 12. In the MOO problem, the weight of transmission latency is noted as $W_1$ while the weight of privacy entropy is noted as $W_2$. The summation of $W_1$ and $W_2$ is 1.

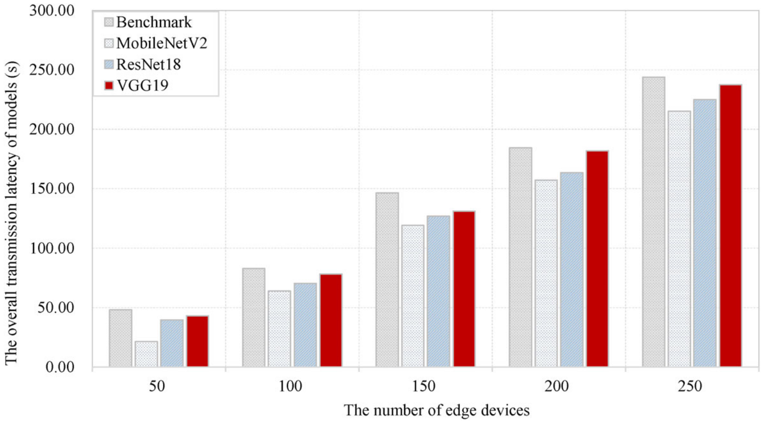


**FIGURE 11** Overall transmission latency of different models transmitted to various edge devices.

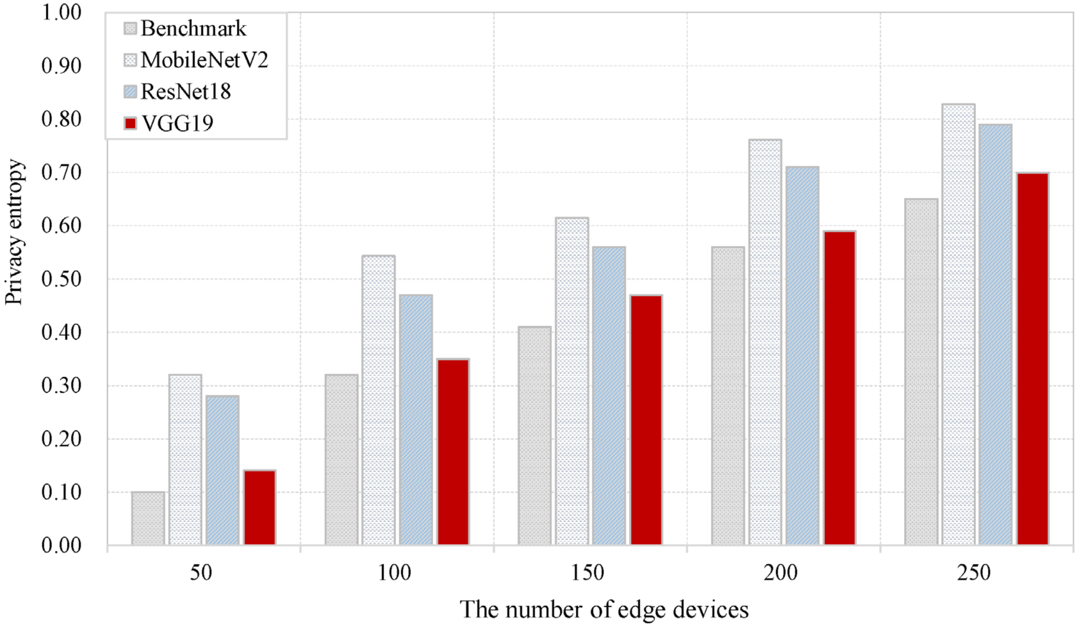


**FIGURE 12** The privacy entropy of various models with different edge devices amount

## 7 | CONCLUSION AND FUTURE WORK

Currently, the outbreak of COVID-19 seriously threatens the individuals' lives. Though many approaches have been proposed to deal with the pandemic, the limitation of computing resources keeps being a conundrum. In this article, a COVID-19 detection method which acts through classifying CXR images in edge computing was proposed. First, the classification of CXR images was implemented by a lightweight model named MobileNet because the model is able to work on edge devices commodiously. In order to make up for the insufficient accuracy of MobileNet, DCGAN was utilized in this article to expand the size of the dataset, thereby improving the accuracy of the MoblieNet. Besides, a MOO problem was solved for offloading the models to edge devices in a way that protects data privacy and reduces information transmission latency at the same time.

According to the accomplished achievement of this article, we will try to extend research results to actual application scenarios. Furthermore, we hope to realize the regular optimization and update of the model by only transmitting a few processed parameters.

## ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant No. 62072251), the Natural Science Foundation of Jiangsu Province (Grant No. BK20211284), and the PAPD fund.

## CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

## DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

## ORCID

*Beijing Chen* https://orcid.org/0000-0002-2506-0427

## REFERENCES

1. Mehta P, McAuley DF, Brown M, et al. COVID-19: consider cytokine storm syndromes and immunosuppression. *Lancet*. 2020;395(10229):1033.
2. Bedford J, Enria D, Giesecke J, et al. COVID-19: towards controlling of a pandemic. *Lancet*. 2020;395(10229):1015-1018.
3. Li Y, Yao L, Li J, et al. Stability issues of RT-PCR testing of SARS-CoV-2 for hospitalized patients clinically diagnosed with COVID-19. *J Med Virol*. 2020;92(7):903-908.
4. Tahamtan A, Ardebili A. Real-time RT-PCR in COVID-19 detection: issues affecting the results. Paper presented at: Expert review of molecular diagnostics; April 2020: 453-454.
5. Wong HYF, Lam HYS, Fong AHT, et al. Frequency and distribution of chest radiographic findings in COVID-19 positive patients. *Radiology*. 2020;296(2):E72-E78.
6. Zhang J, Xie Y, Pang G, et al. Viral pneumonia screening on chest X-rays using confidence-aware anomaly detection. *IEEE Trans Med Imaging*. 2020;40(3):879-890.
7. Wang L, Lin ZQ, Wong A. Covid-net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. *Sci Rep*. 2020;10(1):1-12.
8. Gilbert M, Pullano G, Pinotti F, et al. Preparedness and vulnerability of African countries against importations of COVID-19: a modelling study. *Lancet*. 2020;395(10227):871-877.
9. Chen M, Shi X, Zhang Y, Wu D, Guizani M. Deep feature learning for medical image analysis with convolutional autoencoder neural network. *IEEE Trans Big Data*. 2017;7(4):750-758.
10. Fan DP, Zhou T, Ji GP, et al. Inf-net: automatic COVID-19 lung infection segmentation from CT images. *IEEE Trans Med Imaging*. 2020;39(8):2626-2637.
11. Liu C, Cao Y, Alcantara M, et al. TX-CNN: detecting tuberculosis in chest X-ray images using convolutional neural network. Paper presented at: 2017 IEEE international conference on image processing (ICIP); September 2017:2314-2318; IEEE.
12. Bullock J, Cuesta-Lazaro C, Quera-Bofarull A. XNet: a convolutional neural network (CNN) implementation for medical X-ray image segmentation suitable for small datasets. Paper presented at : Medical imaging 2019: Biomedical Applications in Molecular, Structural, and Functional Imaging; March 2019:453-463; SPIE.
13. Islam MZ, Islam MM, Asraf A. A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images. *Inform Med Unlocked*. 2020;20:100412.

14. Labhane G, Pansare R, Maheshwari S, Tiwari R, Shukla A. Detection of pediatric pneumonia from chest X-ray images using CNN and transfer learning. Paper presented at: 2020 3rd International Conference on Emerging; February 2020:85-92; IEEE.
15. Xu X, Wu Q, Qi L, Dou W, Tsai SB, Bhuiyan MZA. Trust-aware service offloading for video surveillance in edge computing enabled internet of vehicles. *IEEE Trans Intell Transp Syst*. 2021;22(3):1787-1796. doi:10.1109/tits.2020.2995622
16. Hegde C, Jiang Z, Suresha PB, et al. Autotriage-an open source edge computing raspberry pi-based clinical screening system. medRxiv; 2020.
17. Al-Zinati M, Alrashdan R, Al-Duwairi B, Aloqaily M. A re-organizing biosurveillance framework based on fog and mobile edge computing. *Multimed Tools Appl*. 2021;80(11):16805-16825.
18. Whaiduzzaman M, Hossain M, Shovon AR, et al. A privacy-preserving mobile and fog computing framework to trace and prevent COVID-19 community transmission. *IEEE J Biomed Health Inform*. 2020;24(12):3564-3575.
19. Bao J, Chen D, Wen F, Li H, Hua G. CVAE-GAN: fine-grained image generation through asymmetric training. Paper presented at: Proceedings of the IEEE international conference on computer vision; October2017:2745-2754; IEEE.
20. Han C, Hayashi H, Rundo L, et al. GAN-based synthetic brain MR image generation. Paper presented at: IEEE 15th international symposium on biomedical imaging; April 2018:734-738; IEEE.
21. Zhai M, Chen L, Tung F, He J, Nawhal M, Mori G. Lifelong GAN: continual learning for conditional image generatio. Paper preseneted at : Proceedings of the IEEE/CVF International Conference on Computer Vision; October 2019:2759-2768; IEEE.
22. Rajesh S, Paul V, Menon VG, Jacob S, Vinod P. Secure brain-to-brain communication with edge computing for assisting post-stroke paralyzed patients. *IEEE Internet Things J*. 2020;7(4):2531-2538. doi:10.1109/jiot.2019.2951405
23. Kou H, Liu H, Duan Y, et al. Building trust/distrust relationships on signed social service network through privacy-aware link prediction process. *Appl Soft Comput*. 2021;100:106942. doi:10.1016/j.asoc.2020.106942
24. Qi L, Hu C, Zhang X, et al. Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment. *IEEE Trans Ind Inform*. 2021;17(6):4159-4167. doi:10.1109/tii.2020.3012157
25. Wang L, Zhang X, Wang T, et al. Diversified and scalable service recommendation with accuracy guarantee. *IEEE Trans Comput Soc Syst*. 2020;8:1182-1193. doi:10.1109/tcss.2020.3007812
26. Wang L, Zhang X, Wang R, Yan C, Kou H, Qi L. Diversified service recommendation with high accuracy and efficiency. *Knowl Based Syst*. 2020;204:106196. doi:10.1016/j.knosys.2020.106196
27. Yu Y, Gong Z, Zhong P, Shan J. Unsupervised representation learning with deep convolutional neural network for remote sensing images. Paper presented at: International conference on image and graphics; Janruary 2017:97-108; Springer.
28. Liang X, Hu Z, Zhang H, Gan C, Xing EP. Recurrent topic-transition GAN for visual paragraph generation. Paper presented at: Proceedings of the IEEE international confertence on computer vision; October 2017:3362-3371; IEEE.
29. Li Q, Qu H, Liu Z, et al. AF-DCGAN: amplitude feature deep convolutional GAN for fingerprint construction in indoor localization systems. *IEEE Trans Emerg Top Comput Intell*. 2019;5(3):468-480.
30. Kim DD, Shahid MT, Kim Y, et al. Generating pedestrian training dataset using DCGA. Paper presented at: Proceedings of the 2019 3rd International Conference on Advances in Image Processing; November 2019:1-4.
31. Suárez PL, Sappa AD, Vintimilla BX. Infrared image colorization based on a triplet DCGAN architecture. Paper presented at: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops; July 2017:18-23; IEEE.
32. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. Mobilenetv2: Inverted residuals and linear bottlenecks. Paper presented at: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; July 2018:4510-4520; IEEE.
33. Wang W, Li Y, Zou T, Wang X, You J, Luo Y. A novel image classification approach via dense-MobileNet models. *Mob Inf Syst*. 2020;2020:8.

34. Genc S, Akpinar KN, Karagol S. Automated abnormality classification of chest radiographs using MobileNetV2. Paper presented at: 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications; June 2020:1-4; IEEE.
35. Howard A, Sandler M, Chu G, et al. Searching for mobilenetv3. Paper presented at: Proceedings of the IEEE/CVF International Conference on Computer Vision; October 2019: 1314-1324; IEEE.
36. Yang TJ, Howard A, Chen B, et al. Netadapt: platform-aware neural network adaptation for mobile applications. Paper presented at: Proceedings of the European Conference on Computer Vision (ECCV); September 2018: 285-300.
37. Azzouz R, Bechikh S, Said LB. Dynamic multi-objective optimization using evolutionary algorithms: a survey. Paper presented at: Recent advances in evolutionary multi-objective optimization; January 2017: 31-70; Springer.
38. Satpathy S, Mathew S, Suresh V, et al. An all-digital unified static/dynamic entropy generator featuring self-calibrating hierarchical Von Neumann extraction for secure privacy-preserving mutual authentication in IoT mote platforms. Paper presented at: 2018 IEEE Symposium on VLSI Circuits; June 2018: 169-170; IEEE.
39. Rajesh S, Paul V, Menon V, Khosravi M. A secure and efficient lightweight symmetric encryption scheme for transfer of text files between embedded IoT devices. *Symmetry*. 2019;11(2):293. doi:10.3390/sym11020293
40. Vinoj PG, Jacob S, Menon VG, Rajesh S, Khosravi MR. Brain-controlled adaptive lower limb exoskeleton for rehabilitation of post-stroke paralyzed. *IEEE Access*. 2019;7:132628-132648. doi:10.1109/access.2019.2921375
41. Hao Y, Ni Q, Li H, Hou S. Robust multi-objective optimization for EE-SE tradeoff in D2D communications underlaying heterogeneous networks. *IEEE Trans Commun*. 2018;66(10):4936-4949.
42. Jacob S, Menon VG, Al-Turjman F, Vinoj PG, Mostarda L. Artificial muscle intelligence system with deep learning for post-stroke assistance and rehabilitation. *IEEE Access*. 2019;7:133463-133473. doi:10.1109/access.2019.2941491

**How to cite this article:** Xu W, Chen B, Shi H, Tian H, Xu X. Real-time COVID-19 detection over chest x-ray images in edge computing. *Computational Intelligence*. 2022;1-22. doi: 10.1111/coin.12528