Applied Soft Computing 130 (2022) 109656

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

CodnNet: A lightweight CNN architecture for detection of COVID-19 infection

Jingdong Yang [a,1,*], Lei Zhang [a,1], Xinjun Tang [b,*], Man Han [c,*]

[a] School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, China
[b] Department of respiratory and critical care medicine, Zhongshan Hospital, Fudan University, Shanghai, 200032, China
[c] Division of Rheumatology, Guang'anmen Hospital, China Academy of Chinese Medical Sciences, Beijing 100053, China

ARTICLE INFO

Article history:
Received 8 March 2021
Received in revised form 17 August 2022
Accepted 20 September 2022
Available online 24 September 2022

Keywords:
DenseNet
Depthwise separable convolution
Lightweight CNN
COVID-19

ABSTRACT

The application of Convolutional Neural Network (CNN) on the detection of COVID-19 infection has yielded favorable results. However, with excessive model parameters, the CNN detection of COVID-19 is low in recall, highly complex in computation. In this paper, a novel lightweight CNN model, CodnNet is proposed for quick detection of COVID-19 infection. CodnNet builds a more effective dense connections based on DenseNet network to make features highly reusable and enhances interactivity of local and global features. It also uses depthwise separable convolution with large convolution kernels instead of traditional convolution to improve the range of receptive field and enhances classification performance while reducing model complexity. The 5-Fold cross validation results on Kaggle's COVID-19 Dataset showed that CodnNet has an average precision of 97.9%, recall of 97.4%, F1score of 97.7%, accuracy of 98.5%, mAP of 99.3%, and mAUC of 99.7%. Compared to the typical CNNs, CodnNet with fewer parameters and lower computational complexity has achieved better classification accuracy and generalization performance. Therefore, the CodnNet model provides a good reference for quick detection of COVID-19 infection.

1. Introduction

The COVID-19 has become a most devastating pandemic worldwide. Artificial Intelligence is used to accomplish automatic classification of COVID-19 infections on chest X-ray (CXR) images, improves the traditional medical prediction of COVID-19. However, the COVID-19 focal area is characterized by shared space, high variation of infection, and the difference among cases, which brings challenges to the accurate classification of COVID-19.

In 2012, AlexNet won the ImageNet 2012 image recognition challenge [1], showing that CNN had greater advantages in image classification [2]. Subsequently, CNNs developed rapidly and a series of new network structures emerged. Representative networks: VGGNet [3], ResNet [4], DenseNet [5], MobileNet [6], and EffectionNet [7], focus on different aspects: accuracy, efficiency, and scalability etc. Multiple reusable and efficient design principles were also proposed. CNN has achieved good results in the medical field in addition to its amazing performance in the ImageNet 2012 competition. Using various radiology medical datasets, CNN can efficiently implement tasks such as disease classification and lesion tissue segmentation, reducing the work pressure of professional radiologists and improving the efficiency of lesion detection.

DenseNet, as a representative CNN network, uses dense connections to enable shallow features to be passed to deep convolutional layers, making features highly reusable, and thus is widely used in medical fields. X. Luo et al. [8] constructed a new DenseNet structure named UC-DenseNet. It uses DenseNet to extract colonoscopy features of image, which are input to the recurrent neural network and the efficient attention mechanism network (EAM-Net), combining the channel attention module and the spatial attention module, respectively for automatic classification. It enjoys an accuracy improvement of 0.5% to 2% compared to the DenseNet201 model. However, DenseNet201 has multiple model parameters and its classification of large-scale medical images is time-consuming. In contrast, MobileNet is a lightweight CNN structure that uses depthwise separable convolution instead of traditional convolution, significantly reducing the number of model parameters and complexity. C. Wang et al. [9] improved MobileNet by extracting image features using correlation (amplification factor and binary classification probability), ensuring good prediction performance while greatly reducing the model parameters and computational effort. Experiments on the BreakHis dataset show that this network has high recognition performance and computational utilization.

* Corresponding authors.
E-mail addresses: eerfriend@yeah.net (J. Yang), tang.xinjun@zs-hospital.sh.cn (X. Tang), hanman111@126.com (M. Han).
[1] Co-first authors, contributed equally to this work

https://doi.org/10.1016/j.asoc.2022.109656
1568-4946/© 2022 Elsevier B.V. All rights reserved.

In this paper, a novel CNN model, CodnNet is proposed. CodnNet adopts the dense connections of DenseNet to extract multi-scale in-depth features, and the depthwise separable convolution of MobileNet to replace the traditional convolution, which reduces the model complexity and accomplishes automatic COVID-19 detection. Compared with several typical CNN models, the CodnNet model, with fewer parameters and lower complexity, has achieved better classification performance. The contributions of this paper are as follows.

(1) Adding Focus layer and modifying the pooling layer to make sure that the initial image and all feature maps can be transformed to all convolutional layers via dense connections to improve feature utilization.

(2) An efficient depthwise separable convolution of a large convolution kernel is used instead of the traditional convolution to increase the receptive field range and improve the classification performance while reducing the model complexity and the number of parameters.

(3) With lower algorithm complexity and fewer parameters, the proposed model can significantly save bandwidth and reduce costs of storage for large datasets.

2. Related works

To combat the global COVID-19 pandemic, a growing number of researchers worldwide are using CNN models to detect COVID-19. Various state-of-the-art CNN models applied to COVID-19 detection are listed in Table 1. C. Ouchicha et al. [10] constructed a CvdNet model based on ResNet. It extracted COVID-19 early features with different receptive fields and enjoyed good classification performance. However, the use of larger convolution kernels increased the model complexity and the number of parameters. M. Nour et al. [11] used machine learning instead of fully connected layers to improve COVID-19 classification accuracy, firstly using CNN to extract features, K-Nearest Neighbor (KNN), support vector machine (SVM) and decision tree model to classify, and finally using Bayesian algorithm to optimize parameters. Experiments showed that SVM classification performance is the best. However, multi-stage prediction introduces a large error rate. N. Chowdhury et al. [12] proposed a PDCOVIDNet model using dilated convolution instead of traditional convolution and constructed a parallel feature transform channel with traditional convolution, which effectively improved the COVID-19 classification accuracy, but the superposition property of dilated convolution led to some detailed features loss. T. Mahmud et al. [13] proposed the CovXNET model, which used different expansion rates to extract in-depth features of COVID-19 CXR images and constructed an integrated model with multi-network overlay to improve classification performance. The model has good results for small size samples, but the running time increases exponentially when the size of samples increases.

M. Heidari et al. [14] used two image preprocessing methods (noise filtering and contrast normalization), followed by the classification of CNN model. The model led to an improvement in classification accuracy, but the preprocessing methods resulted in the loss of some important features. H. Panwar et al. [15] used the transfer learning framework VGG-19 and obtained pre-training weights on the ImageNet dataset and validated it against a small number of COVID-19 samples and achieved better classification results, but required a longer pre-training time.

R. Karthik et al. [16] constructed a CNN model with feature filtering learning that used a quadratic loss function as a convolutional filter for each label-specific category, which better improves the classification performance. However, the model has a high algorithm complexity and the trade-off between classification performance and time cost needs to be considered. K. Shibly et al. [17] used a combined VGG16 and Faster R-CNN model to predict COVID-19 with better classification performance compared to the traditional ResNet-50 model, which first extracted multiple object frames using the Faster R-CNN model and then classified the features in the object frames. However, the model has a high algorithm complexity. V. Arora et al. [18] tested against COVID-CT Scan and SARS-COV-2 CT-Scan sample sets using pre-training models such as XceptionNet, MobileNet, InceptionV3, DenseNet, ResNet50 and VGG 16. The experimental results show that MobileNet has the best performance. However, the model is not suitable for all types of lesion prediction.L. Wang et al. [19] proposed a new model, COVID-NET, which used reinforcement learning to construct a lightweight structure, PEPX, to reduce the algorithm complexity, and the model worked well for specific COVID-19 detection. However, the classification performance was likely to degrade when the sample distribution changed. A. Khan et al. [20] constructed a CoroNet model based on Xception frame, which used depthwise separable convolution instead of traditional convolution to reduce the algorithm complexity and number of parameters. However, the classification performance decreased.

G. Jia et al. [21] proposed an improved MobileNet model applied to CXR images for a 5-classification study of COVID-19, tuberculosis, viral pneumonia, bacterial pneumonia, and normal controls, and an improved ResNet model applied to CT images for a 3-classification study of COVID-19, non-COVID-19 infections, and normal controls. There is still room for improvement in the complexity of the proposed lightweight model.

It can be seen that with respect to the classification of COVID-19, the less complex model can also show good performance compared to the multi-stage models and the more complex CNNs.

3. Materials and methods

3.1. Dataset description

The COVID-19 data is publicly available on Kaggle's website [22–24] and it is collected from different databases: chest X-ray images with COVID-19 are taken from the Italian Society of Medical and Interventional Radiology COVID-19 Database (SIRM) [25] and from Novel Corona Virus 2019 Dataset which is developed by Cohen et al. in Github [26], as well as from different recently published articles. Viral pneumonia and normal images are collected from Kaggle's Chest X-ray pneumonia dataset [27]. The data distribution is shown in Table 1, where 10 611 images are used as the training set, 3028 images as the validation set, and 1514 images as the test set. Fig. 1 shows some images of this dataset (see Table 2).

3.2. Evaluation index and data augmentation

The evaluation indexes include Accuracy, Precision [28], Recall [29], F1score [30], area under P–R curve (AP) [31], and area under ROC curve (AUC) [32,33]. TP, FP, TN and FN were true positive, false positive, true negative, and false negative, respectively [34]. The formulas are as follows.

$$Accuracy = (TP + TN)/(TN + FN + TP + FP) \tag{1}$$

$$Precision = TP/(TP + FP) \tag{2}$$

$$Recall = TP/(TP + FN) \tag{3}$$

$$F1score = (2 \times Pre \times Rec)/(Pre + Rec) \tag{4}$$

Accuracy can determine the total correct rate, but in the case of unbalanced samples, it is not the optimal indicator to measure

Table 1
Representative works for CXR images based on the detection of COVID-19 Infection.

Author	Method	Dataset	Result
C. Ouchicha et al. [10]	CvdNet	Kaggle's COVID-19 Radiography Database (219 COVID-19 positive, 1341 normal and 1345 viral pneumonia chest × -ray images)	Average accuracy of 97.20% for detecting COVID-19 and an average accuracy of 96.69% for three-class classification
M. Nour et al. [11]	A Novel Medical Diagnosis model based on Deep Features and Bayesian Optimization	Kaggle's COVID-19 Radiography Database	Accuracy of 98.97%, recall of 89.39%, specificity of 99.75%, and F-score of 96.72%
N. Chowdhury et al. [12]	A Parallel-Dilated Convolutional Neural Network Architecture	Kaggle's COVID-19 Radiography Database	Accuracy, precision, recall, and F1 scores reach 96.58%, 96.58%, 96.59% and 96.58%
T. Mahmud et al. [13]	CovxNet	1583normal X-rays, 1493 non-COVID viral pneumonia X-rays and 2780 bacterial pneumonia X-rays	Accuracy of 97.4% for COVID/Normal, 96.9% for COVID/Viral pneumonia, 94.7% for COVID/Bacterial pneumonia, and 90.2% for multiclass COVID/normal/Viral/Bacterial pneumonias
M. Heidari et al. [14]	Chest X-ray images with preprocessing algorithms	415 images depict with the confirmed COVID-19 disease, 5179 with other community-acquired non-COVID-19 infected pneumonia, and 2880 normal (non-pneumonia) cases	Accuracy of 94.5% (2404/2544) with a 95% confidence interval of [0.93,0.96] in classifying 3 classes 98.4% recall (124/126) and 98.0% specificity (2371/2418) in classifying cases with and without COVID-19 infection
H. Panwar et al. [15]	VGG	673 radiology images of 342 unique patients	Recall and specificity of the proposed model is 76.19%, and 97.22%
R. Karthik et al. [16]	Channel-shuffled dual-branched CNN	558 COVID-19 Chest X-rays	F1score of 97.20% and an accuracy of 99.80% on the COVID-19 X-ray set
K. Shibly et al. [17]	COVID faster R-CNN	COVID chest X-ray dataset curated by Dr. Joseph Cohen	Accuracy of 97.36%, recall of 97.65%, and a precision of 99.28%
V. Arora et al. [18]	MobileNet	The COVID-CT-Dataset contained 349 COVID-19 CT images, and 463 non-COVID-19 CTs. In SARS-COV-2 CT, of the total 2482 images, the non-COVID-19 subjects accounted for 1230 CT scans.	The recall of 96.11% and 100% respectively; precision of 96.11% and 100% respectively; F-1 scores of 96.11% and 100% respectively; and accuracy of 94.12% and 100% respectively.
L. Wang et al. [19]	COVID-NET	358 CXR images	Accuracy of 93.3%, recall of 91.0%
A. Khan et al. [20]	CoroNet	290 COVID-19 chest Radiography images, 1203 normal, 660 bacterial Pneumonia and 931 viral Pneumonia cases	Accuracy of 89.6% and precision and recall rate for COVID-19 cases are 93% and 98.2%
G. Jia et al. [21]	MobileNet, ResNet	1170 COVID-19 CXR images, COVIDx-CT dataset which consists of 143,778 training images	Accuracy of 99.6% on the five-category CXR image dataset and accuracy of 99.3% on the CT image dataset

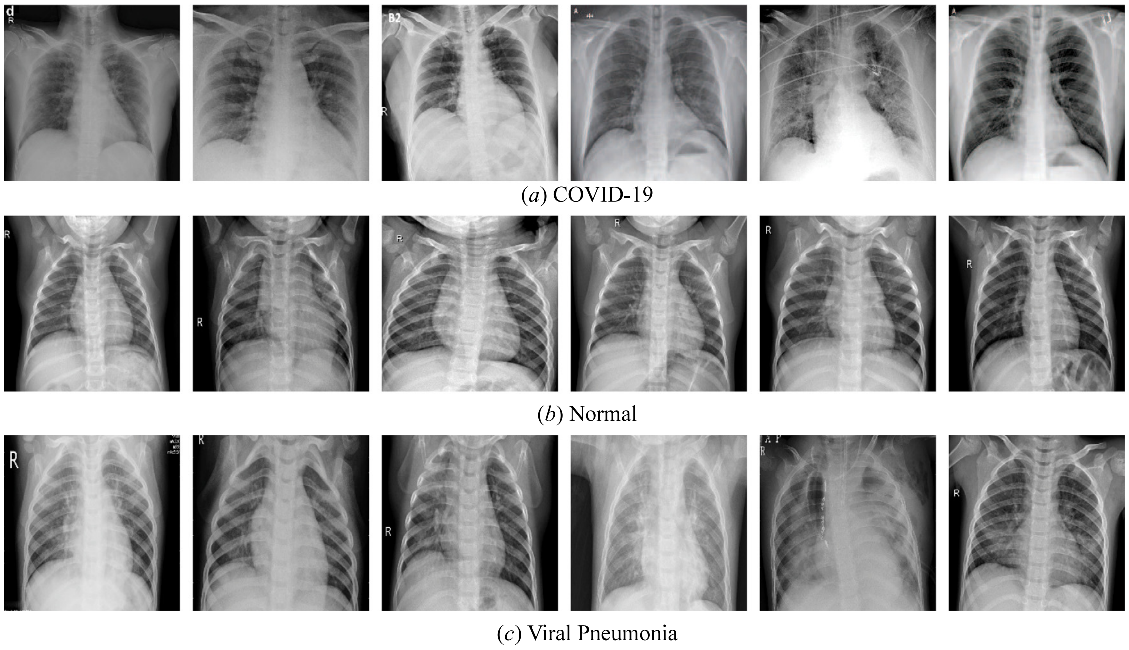

Fig. 1. Kaggle's COVID-19 radiography database.

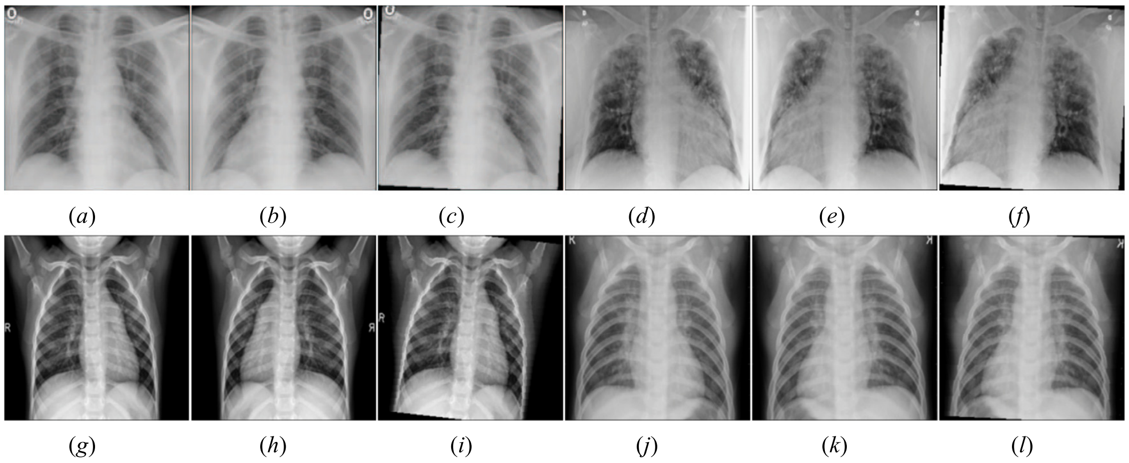

Fig. 2. Data augmentation where $(a)(d)$ refers to original image of COVID-19, (g) is original image of Normal, (j) is original image of Viral Pneumonia, $(b)(e)(h)(k)$ refers to randomly horizontal flip, and $(c)(f)(i)(l)$ refers to randomly rotation in range of $(-10°, +10°)$ respectively.

Table 2
The distribution of Kaggle's COVID-19 radiography database.

Class	# of samples	Train	Valid	Test
COVID-19	3616	2533	361	722
NORMAL	10 192	7135	1019	2038
Viral Pneumonia	1345	943	134	268
Total	15 153	10 611	1514	3028

the results; precision represents the accuracy of prediction in positive samples; recall means the probability of predicted as positive in the actual positive samples; F1score can represent a balance of precision and recall; AUC reflects the generalization performance of the model; AP represents the precision of the detection, which shows the classification performance in uneven multi-class samples.

Since the sample size of COVID-19 is small in the dataset, data enhancement [35] is needed to improve model performance. The CXR images in the dataset have high pixel and will not produce too many noisy points. Therefore, only a simple transform is made for this dataset to keep the original features as much as possible. As shown in Fig. 2, we reduce the resolution of the CXR image to 256 × 256 pixels, apply random horizontal flips and random horizontal rotations in the interval of $(-10°, +10°)$, and perform normalization operations, so that the normalized RGB mean and standard deviation are 0.485, 0.456, 0.406 and 0.229, 0.224, 0.225, respectively.

3.3. The CodnNet model

In COVID-19 CXR images, the early features of frosted glass turbidity can be found, which have different sizes and positions. Therefore, more efficient feature delivery methods are needed to enable features under different receptive fields to fuse with each other. Therefore, we propose a lightweight DenseNet model, CodnNet, which is characteristic of lower complexity and higher propagation efficiency based on the idea of dense connection and depthwise separable convolution.

We use the Focus layer [36] in the initial feature extraction layer of the model to resize the original image. As shown in Fig. 3, a 1 × 4 × 4 image through the Focus layer will become a 4 × 2 × 2 image. That is, the number of channels is multiplied by 4 and the width and height are reduced by half. The Focus layer converts the information in the *w-h* plane to the channel dimension and then extracts different features via the convolution layer, and this approach can effectively reduce the information

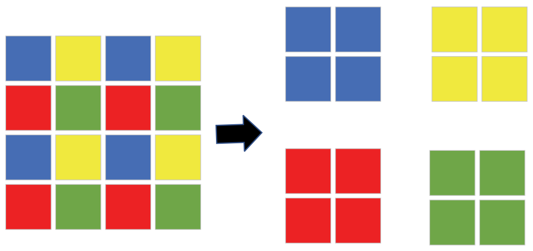

Fig. 3. Focus layer.

loss due to downsampling. As shown in Fig. 4, in the DenseNet, the input image passes through a 7 × 7 convolutional layer with a stride of 2. The number of channels becomes I, and the width and height are reduced by half. Then a 3 × 3 maxpooling layer with a stride of 2 is used to reduce the width and height by half, but the original image features cannot pass to the feature extractor afterwards because no connection is used.

Therefore, we add a dense connection to pass the feature map. The original image will pass through a 3 × 3 convolutional layer with a stride of 2, with the channel number becoming I. After a Focus layer, the channel number is 12, and after the concatenation operation, the feature map with channel number I+12 is finally generated. Then a 3 × 3 depthwise separable convolutional layer with a stride of 2 is used to adjust the feature map size, and a maximum pooling layer is added to the dense connection. The final output is a feature map with 2I+24 channels. In the initial feature extraction layer, we use a small convolutional kernel instead of a large convolutional kernel to reduce the complexity of the model, and add an additional layer of depthwise separable convolution to maintain the same receptive field. In the initial feature map, all pixel points of the initial image can be passed to the feature extractor and even to the final classifier due to the addition of skip connections in the Focus layer combined with DenseNet dense connections.

As shown in Fig. 5, we improve the convolution layer in DenseBlock by using depthwise separable convolution instead of the traditional convolution layer, which includes depthwise convolution and pointwise convolution. In the DenseBlock, the number of channels of feature map is generated to *n* × *growth rate* using 1 × 1 convolution layer, and then the number of channels is changed to *growth rate* using 3 × 3 convolution layer. We keep the first layer of 1 × 1 convolution layer, replace the original 3 × 3 convolution layer with 7 × 7 depthwise

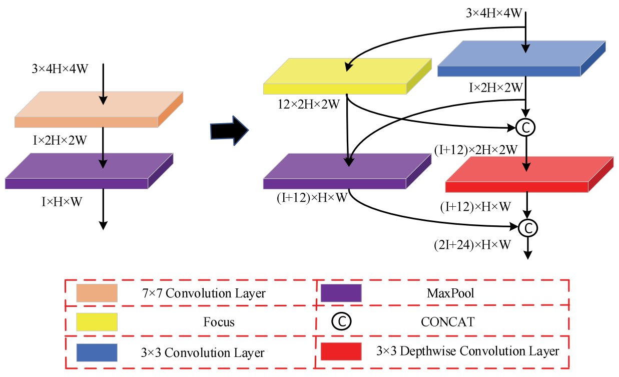

Fig. 4. Improvement of initial feature extraction layer.

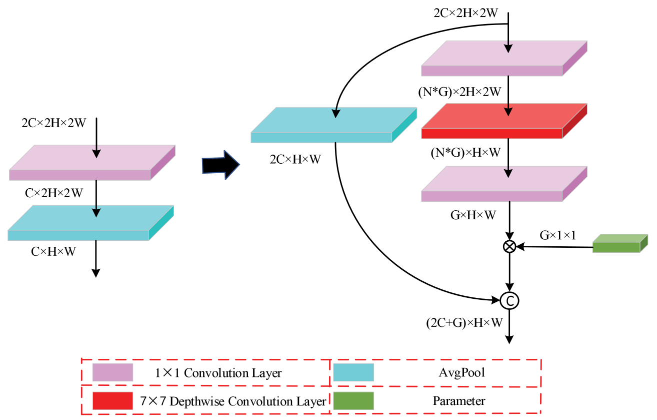

Fig. 6. Improvement of transition layer.

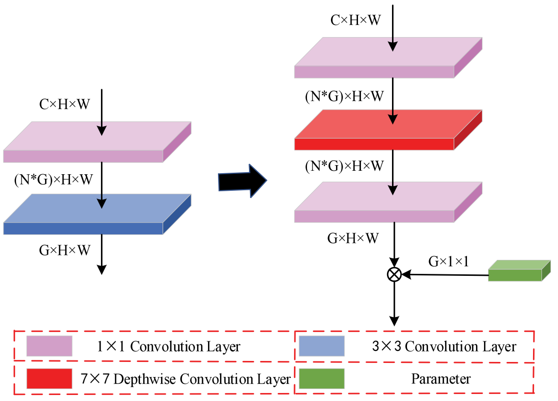

Fig. 5. Improvement of DenseBlock convolutional layer.

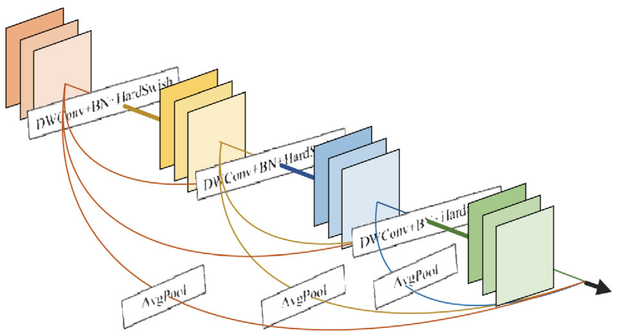

Fig. 7. The diagram of the CodnBlock architecture.

convolution layer and 1 × 1 convolution layer, and finally use a set of learnable weights to multiply with the feature map to make the feature map focus on more useful channels. Meanwhile, we use HardSwish nonlinear transformation [37] instead of ReLU in the convolutional layer. These improvements can improve the classification performance of the original model effectively and significantly reduce the number of operations and model parameters.

As shown in Fig. 6, we modify the transition layer in DenseNet so that it can be incorporated into DenseBlock. The original transition layer is structured as follows. The output of DenseBlock passes through a 1 × 1 convolutional layer with the number of feature map channels halved, then it passes an average pooling layer with the feature map size halved. This transition layer follows the DenseBlock and is intended to reduce feature map size and number of channels. However, when the number of channels is small, the 1 × 1 convolution layer cannot effectively reduce the number of parameters, but instead degrades the model performance. Therefore, we eliminate the 1 × 1 convolutional layer and add a depthwise separable convolutional layer with a stride of 2 to the DenseBlock, and add an average pooling layer to the dense connection, which can effectively prevent the information loss caused by utilizing the maximum pooling layer alone.

By modifying the convolutional and transition layers, the CodnBlock structure is shown in Fig. 7. Assuming a CodnBlock with l layers, the depthwise convolutional layer of the first *l-1* layers has a stride of 1, and the depthwise convolutional layer of the last layer has a stride of 2 to act as a transition layer, and the feature map size is reduced by half. If the input image is C × 2H × 2W, the output of feature map is (C+l*growth rate*) × H × W. The output of CodnBlock can be expressed as follows.

$$x_l = [AVG\left(H\left([x_1, x_2, \ldots, x_{l-2}]\right)\right), H\left(x_{l-1}\right)] \tag{5}$$

where x_l is the output of layer l, $[x_1, x_2, \ldots, x_{l-2}]$ refers to the concatenation of the lth layer to *(l-2)*th layer output, H is a compound function of three continuous operations, including BN layer, HardSwish layer, and convolutional layer, and AVG is the average pooling layer.

As shown in Fig. 8, the input size of 256 × 256 image is changed to 4 copies of 128 × 128 by the Focus layer of the initial feature extraction layer, then the image size is changed to 64 × 64 by the maxpooling layer, and finally pass to the final classifier by the average pooling layer in several CodnBlock layers. With the improved dense connection, both the input image and each feature map layer can pass the original feature information to all subsequent convolutional layers, and the feature map can be efficiently utilized compared to the original DenseNet.

The CodnNet model consists of 4 CodnBlocks as shown in Fig. 9. The size of input image is 3 × 256 × 256, the initial number of feature channels is 16. Then after the initial feature extraction, the feature map becomes 56 × 64 × 64. The channel multiplier in CodnBlock is 4, and the growth rate is 32. The first 3 layers of CodnBlock fuse the transition layer. After the first layer of CodnBlock, the feature map becomes 88 × 32 × 32; after the second layer of CodnBlock, the feature map becomes 120 × 16 × 16; after the third layer of CodnBlock, the feature map becomes 152 × 8 × 8; after the fourth layer of CodnBlock, the feature map becomes 184 × 8 × 8. After the feature extraction, the feature map is flattened by global avgpool so that the model can accept input images of different sizes. The classifier is two fully connected layers. Because the number of feature extraction layers is small, the classifier input includes only 152 parameters, and

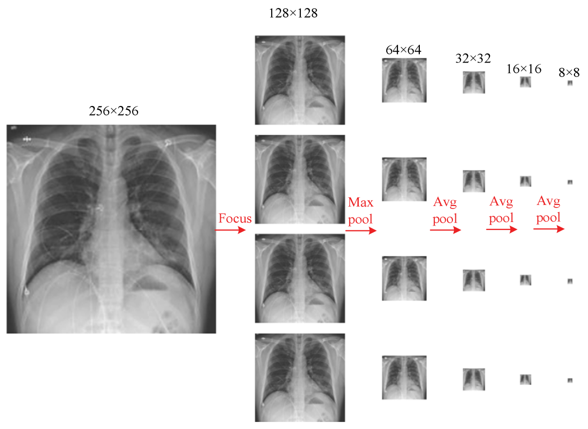

Fig. 8. The change of input image.

to increase the classification performance, the classifier parameters are expanded to 512 and then mapped to the classification number. The detailed structure of the model is as follows.

(A) Convolutional layer: Convolution is a basic operation of the convolutional neural network, which can automatically extract features. Convolution is a mathematical operation, which uses a series of filters to generate a new set of feature maps through a sliding window in inputs. CodnBlock uses a traditional convolution layer with a convolution kernel of 1 × 1 to fuse features, and a depthwise convolution with a larger convolution kernel to extract features, reducing the number of model parameters and complexity. The equations of traditional convolution and depthwise convolution are as follows.

$$Z^{(l)}(c,i,j)=\sum_{d}\sum_{m}\sum_{n}x^{(l-1)}(d,i+m,j+n)\,k^{(l)}(m,n) \tag{6}$$

$$Z^{(l)*}(c,i,j)=\sum_{m}\sum_{n}x^{(l-1)}(c,i+m,j+n)\,k^{(l)}(m,n) \tag{7}$$

where $Z^{(l)}$ is the traditional convolutional layer output of the lth layer, $Z^{(l)*}$ is the depthwise convolutional layer output of the lth layer, c is the number of channels of the output feature map, $x^{(l-1)}$ is the input of the lth layer, $k^{(l)}$ is the lth filter of size $m \times n$.

(B) Activation function: after the convolution, the nonlinear transformation of the feature graph is required to complete the activation operation. ReLU [38] is adopted in initial feature extraction layer. Compared with other activation functions, when the input of ReLU is greater than 0, the gradient will not disappear and the calculation speed is faster. But when the input is less than 0, the gradient disappears. The formula of ReLU is as follows.

$$ReLU(x)=\begin{cases}x, x>0\\0, x\le 0\end{cases} \tag{8}$$

where x is the input of ReLU, $g(x)$ is the output of ReLU.

HardSwish activation function is used in the later layers. Compared with ReLU, HardSwish can significantly improve the network classification accuracy although delay may increase. Compared with other activation functions, HardSwish can not only reduce the error of shallow layers in quantization mode, but also effectively extract in-depth features. Therefore, we only use it in CodnBlock and the classifier. The formula of HardSwish is as follows.

$$ReLU6(x)=\begin{cases}6, x\ge 6\\x, x>0\\0, x\le 0\end{cases} \tag{9}$$

$$HardSwish(x)=x*\frac{ReLU6(x+3)}{6} \tag{10}$$

where x is the input, $ReLU6(x)$ is the output of ReLU6, $HardSwish(x)$ is the output of HardSwish.

(C) Pooling layer: after the convolutional layer, the size of the feature map is still large. The addition of a pooling layer can reduce the size of the feature map, thus reducing the number of fully connected layer parameters. It can prevent overfitting and enable the feature graph to have invariance of translation and rotation. To reduce the useless information in the feature map, the max pooling layer is used in the extraction of shallow features in this paper. After CodnBlock, the average pooling layer is used to retain the feature information as much as possible.

(D) Batch normalization layer: it is a technique used to optimize models during training [39]. It calculates the mean and variance of each mini-batch data and then normalizes it. The batch normalization layer can effectively reduce gradient vanishing/explosion phenomenon and accelerate model training speed. The formula of the batch normalization operation is as follows.

$$\mu_B=\frac{1}{n}\sum_{i=1}^{n}x_i \tag{11}$$

$$\sigma_B^2=\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu_B)^2 \tag{12}$$

$$x_i'=\frac{x_i-\mu_B}{\sqrt{\sigma_B^2+\varepsilon}} \tag{13}$$

$$y_i=\gamma_i\cdot x_i'+\beta_i \tag{14}$$

where x_i is the input, n is the total number of features in the input feature map, y_i is the output, μ_B, σ_B^2 is the mean and variance of x, x_i' refers to the normalization of x, γ_i, β_i are model learnable parameters.

(E) Fully connected layer: its essence is a linear transformation from one feature space to another. Map the "distributed feature representation" which is learned in the previous layer to the sample space. It plays the role of "classifier" in CNN. In this paper, after the last fully connected layer, the softmax [40] activation function is used to convert the input into probability. The formula of the softmax activation function is as follows.

$$softmax(z_j)=\frac{e^{z_j}}{\sum_{k=1}^{N}e^{z_k}};j=1,\ldots,N \tag{15}$$

where $z=[z_1,\ldots,z_N]$ is the input, N is the input dimension and $softmax(z_j)\in(0,1)$, $\sum_{j=1}^{N}softmax(z_j)=1$

(F) Dropout layer: in the neural network training, a part of neurons is randomly discarded at a certain probability to simplify the network, which can effectively reduce the high correlation among features and prevent network overfitting. In the testing, the Dropout layer [41] did not work, so a fully connected network was used.

Since COVID-19 CXR images have some typical pathological features, the output feature map of the convolutional layer can be reused to maximize the recall of each convolutional layer to the characteristics of the infected region, and then accurately classify COVID-19.

3.4. The implementation details of the CodnNet model

In this paper, we conduct triple classification automatic detection experiments for Kaggle COVID-19 CXR images. We first divide each class of dataset into 10 equal parts for 5-fold cross-validation [42]. The samples of each fold are processed as follows: 70% of samples are used for the training set, 10% for the validation set, 20% for the test set.

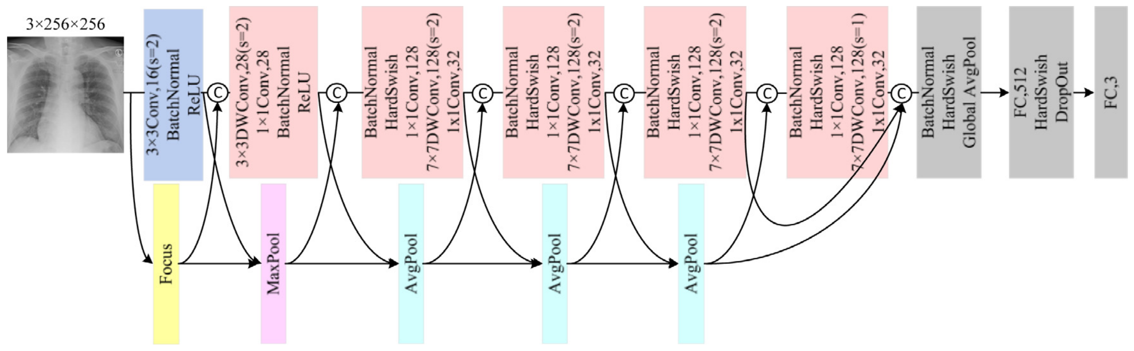
3×256×256
3×3Conv,16(s=2) BatchNormal ReLU
Focus
C
3×3DWConv,28(s=2) 1×1Conv,28 BatchNormal ReLU
MaxPool
C
BatchNormal HardSwish 1×1Conv,128 7×7DWConv,128(s=2) 1x1Conv,32
AvgPool
C
BatchNormal HardSwish 1×1Conv,128 7×7DWConv,128(s=2) 1x1Conv,32
AvgPool
C
BatchNormal HardSwish 1×1Conv,128 7×7DWConv,128(s=2) 1x1Conv,32
AvgPool
C
BatchNormal HardSwish 1×1Conv,128 7×7DWConv,128(s=1) 1x1Conv,32
C
BatchNormal HardSwish Global AvgPool
FC,512 HardSwish DropOut
FC,3

Fig. 9. The architecture of CodnNet model.

Table 3
The implementation details of the CodnNet model.

```
Input   COVID-19 Chest X-Ray images Training set $\delta_1$, Valid set $\delta_2$, Test set $\delta_3$
          $\mu \rightarrow$ the initial learning rate
          $\xi \rightarrow$iteration step
          $\epsilon \rightarrow$the maximum number of iterations
          $\alpha \rightarrow$iteration step in the training
          $\beta \rightarrow$ the number of images covered in one training
          $\eta_1 \rightarrow$the maximum iteration number of one epoch for training set ($\eta \leftarrow \delta_1/\beta$)
          $\eta_2 \rightarrow$the maximum iteration number of one epoch for validation set ($\eta \leftarrow \delta_2/\beta$)
          $\eta_3 \rightarrow$the maximum iteration number of one epoch for testing set ($\eta \leftarrow \delta_3/\beta$)
          $\lambda \rightarrow$variables of saved accuracy ($\lambda \leftarrow 0$)
Output  $\omega \rightarrow$CNN weight
        Start
1.      Initialize the CNN parameters: $\mu$, $\epsilon$, $\beta$, $\lambda$
2.      Light preprocessing the images with 3×256×256 resolution
3.      Train the CNN and computing the weights
4.      for $\xi$ = 1 to $\epsilon$ do
5.          for $\alpha$ = 1 to $\eta_1$ do
6.                  Select a mini-batch from $\delta_1$ with the $\beta$ size
7.                  Forward propagation and compute the loss using $\mu$
8.                  Back-propagation and update $\omega$ with SGD optimization
9.          end
10.         for α = 1 to $\eta_2$ do
11.                 Select a mini-batch from $\delta_2$ with the $\beta$ size
12.                 Forward propagation and get the results of the CNN
13.         end
14.         calculate the valid accuracy from the results
15.         If the accuracy > $\lambda$ do
16.                 Save $\omega$ and $\lambda \leftarrow$the accuracy
17.         end
18.     end
19.     for α = 1 to $\eta_3$ do
20.         Select a mini-batch from $\delta_3$ with the $\beta$ size
21.         Forward propagation and get the results of the CNN
22.     end
23.     calculate the test accuracy from the results
```

A cross-entropy loss function is used in the training to calculate the loss between the output of Softmax and the actual probability. The optimizer adopts SGD, and the learning rate is 1e-3. The iteration step is 100 epoch, and the batch is 32. The detailed implementation process is shown in Table 3.

4. Results

4.1. Ablation study

The factors that affect the complexity of the model mainly include the depth and width of the model. The width factors that affect the complexity of DenseNet are the growth rate and the number of initial feature channels. Therefore, we set the growth rate as 32,16,8, and the number of initial feature channels as 64,32,16 for CodnNet and DenseNet13 respectively, and the results are shown in Table 4.

The analysis shows that the growth rate has the greatest impact on the classification performance in CodnNet and DenseNet13, while the number of initial feature channels has less impact on the classification performance. This is because both CodnNet and DenseNet13 use dense connections to make feature maps highly reusable, and employ the initial feature channels for extraction of shallow features, which have less impact on classification performance. However, the growth rate extracts in-depth features, which have more impact on classification. According to the experimental results, we set the growth rate as 32 and the initial feature channel as 16, which is the best comprehensive classification performance.

In this paper, based on the DenseNet13 ($g = 32$, $I = 16$, $MACs = 0.2$, $Params = 0.17$), the depthwise separable convolution parameters are investigated to test the impact of different scale convolution kernels on classification performance, as shown in Fig. 10. We also test the impact of different size depthwise separable convolution, such as 3×3($MACs = 0.07$, $Params = 0.04$), 5×5($MACs = 0.08$, $Params = 0.05$), 7×7($MACs = 0.09$, $Params = 0.06$), and 9×9 ($MACs = 0.1$, $Params = 0.08$) on

Table 4
Impact of model width on CodnNet and DenseNet13.

Growth rate (g)	Initial feature channels (I)						
		CodnNet			DenseNet13		
		I = 64	I = 32	I = 16	I = 64	I = 32	I = 16
g = 32	Precision	0.977	0.980	0.977	0.975	0.977	0.973
	Recall	0.973	0.969	0.974	0.972	0.964	0.963
	F1score	0.975	0.974	0.975	0.974	0.970	0.968
	Accuracy	0.985	0.984	0.984	0.981	0.979	0.978
g = 16	Precision	0.977	0.976	0.970	0.978	0.965	0.961
	Recall	0.971	0.978	0.967	0.971	0.956	0.953
	F1score	0.974	0.977	0.968	0.975	0.960	0.957
	Accuracy	0.983	0.983	0.980	0.981	0.972	0.970
g = 8	Precision	0.969	0.973	0.943	0.965	0.961	0.953
	Recall	0.973	0.962	0.972	0.962	0.953	0.958
	F1score	0.972	0.967	0.957	0.963	0.957	0.955
	Accuracy	0.979	0.978	0.970	0.972	0.970	0.967

Table 5
Average evaluation metrics of various models for 3-class CXR images.

Models	Precision	Recall	F1score	Accuracy	mAP	mAUC
DenseNet201	0.982 ± 0.003	0.975 ± 0.003	0.978 ± 0.002	0.985 ± 0.002	0.994 ± 0.001	0.997 ± 0.001
DenseNet121	0.979 ± 0.004	0.972 ± 0.005	0.975 ± 0.004	0.983 ± 0.003	0.993 ± 0.001	0.997 ± 0.001
DenseNet13	0.976 ± 0.003	0.970 ± 0.004	0.973 ± 0.003	0.981 ± 0.003	0.992 ± 0.001	0.997 ± 0.001
MobileNetV2	0.976 ± 0.003	0.971 ± 0.003	0.974 ± 0.002	0.982 ± 0.001	0.993 ± 0.001	0.997 ± 0.001
MobileNetV3L	0.978 ± 0.002	0.972 ± 0.005	0.975 ± 0.003	0.983 ± 0.002	0.995 ± 0.001	0.998 ± 0.001
MobileNetV3S	0.976 ± 0.002	0.970 ± 0.005	0.973 ± 0.003	0.981 ± 0.003	0.993 ± 0.001	0.997 ± 0.001
CodnNet	0.979 ± 0.004	0.974 ± 0.003	0.977 ± 0.003	0.985 ± 0.002	0.993 ± 0.001	0.997 ± 0.001

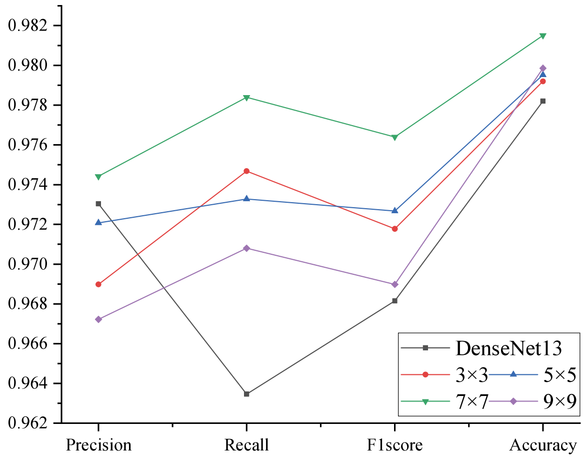

Fig. 10. Impact of different depthwise separable convolution kernels.

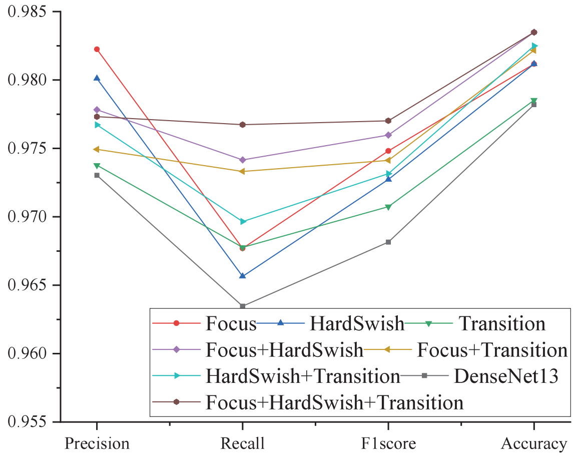

Fig. 11. Impact of each major module.

classification performance. The analysis shows that the classification performance is gradually rising as the scale of depthwise separable convolutional kernel increases, and the model performance reaches the highest when the convolutional kernel is 7 × 7, and decreases when the scale of convolutional kernel is increased again. Therefore, we choose 7 × 7 depthwise separable convolution as the CodnNet convolutional layer.

In this paper, based on DenseNet13 (g = 32, I = 16), the main modules (initial feature extraction layer, transition layer, HardSwish) are ablated experimentally, as shown in Fig. 11. Adding the initial feature extraction layers and HardSwish separately improve the DenseNet13 accuracy by 0.3%. While adding the initial feature extraction layers and HardSwish together increase the accuracy by 0.5%. Improving the transition layer alone achieves less performance improvement because the reduction of the 1 × 1 convolutional layer in the original transition layer makes the model less complex. Regarding experimental results, we choose adding the combination of the initial feature extraction layer (Focus), HardSwish and the transition layer (Transition), which has best classification performance with low algorithm complexity.

Fig. 12 gives the visualization of attention features on COVID-19 infection for five comparative models, including DenseNet13 (g = 32, I = 16) model, and three improved models based on DenseNet13, adding 7 × 7 depthwise separable convolution (DSC), initial feature extraction layer, HardSwish and CodnNet, respectively. The DenseNet13 model uses traditional 3 × 3 convolution layers due to the limited number of layers. The receptive field is small and can only focus on local detailed features. With the addition of 7 × 7 depthwise separable convolution, the receptive field of DenseNet13 is increased. With the addition of initial feature extraction layer, the shallow features of the original image are extracted and a larger range of features with many noise points are also included. HardSwish nonlinear transformation focus on more accurate local features. CodnNet takes into account all advantages of modules, and pays attention to in-depth and shallow features on both sides of lung lesions.

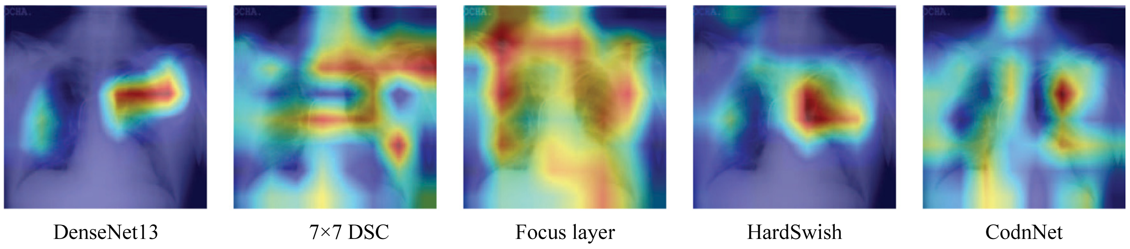

Fig. 12. Visualization of the attention of different module features.

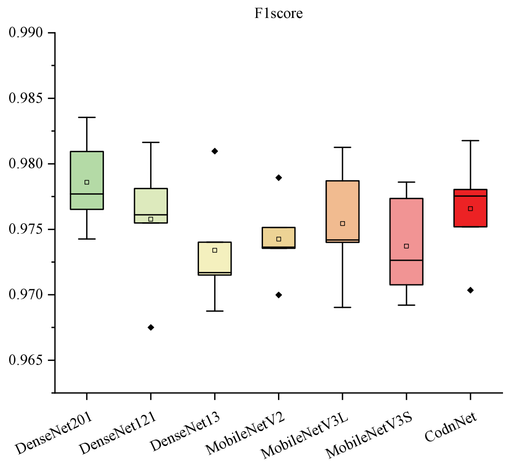

Fig. 13. F1score results for different models.

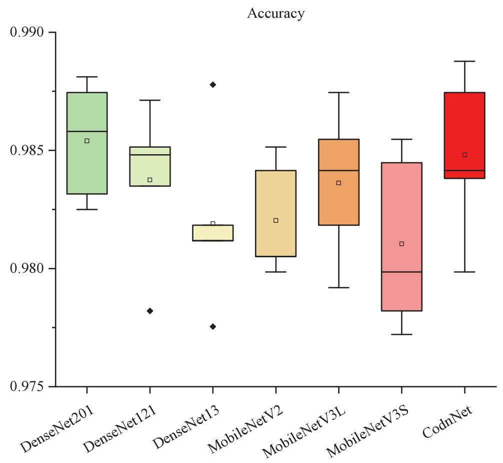

Fig. 14. Accuracy results of different models.

4.2. Evaluation of classification for 3-class COVID-19 datasets

As shown in Table 5, we first conduct 5-fold cross validation test using DenseNet201 (g = 32, I = 64) and DenseNet121 (g = 32, I = 64), with an average accuracy of 98.5% and 98.3%, respectively, and the average accuracy of DenseNet121 is reduced by 0.2% compared to DenseNet201. It can be seen that as the model depth decreases, the detection accuracy of DenseNet framework does not decrease significantly. We continue to reduce the depth of DenseNet so that the number of 3 × 3 convolutional layers of DenseBlock is 1 (DenseNet13(g = 32, I = 64)) with an accuracy of 98.1%. The analysis shows that the DenseNet network can achieve better classification performance with fewer convolutional layers due to its dense connections, which plays a key role in reducing the complexity of the model.

Table 6
Performance of CodnNet after 5-fold Cross-validation for 3-class CXR images.

FOLD	Precision	Recall	F1score	Accuracy	mAP	mAUC
FOLD 1	0.977	0.974	0.975	0.984	0.992	0.997
FOLD 2	0.984	0.973	0.978	0.988	0.994	0.998
FOLD 3	0.980	0.976	0.978	0.984	0.994	0.997
FOLD 4	0.983	0.980	0.982	0.989	0.995	0.999
FOLD 5	0.972	0.969	0.970	0.980	0.989	0.996

Therefore, we construct a lightweight model, CodnNet (g = 32, I = 16) based on DenseNet13(g = 32, I = 16). CodnNet achieves an Accuracy of 98.5% for 3-class classification, which is 0.4% better than DenseNet13(g = 32, I = 64), 0.2% better than DenseNet121, and the same as DenseNet201. The comparison of classification performance is shown in Table 5. The analysis shows that, compared with MobileNetV2 [43], V3L, and V3S, CodnNet achieves a precision increased by 0.3%, 0.1%, and 0.3%, a recall increased by 0.3%, 0.2%, and 0.4%, a F1score increased by 0.3%, 0.2%, and 0.4%, and an accuracy increased by 0.3%, 0.2%, and 0.4%, respectively. It can be seen that the CodnNet model has a higher precision and recall and better classification accuracy and generalization performance.

In clinical medical image classification, F1score and accuracy are two typical evaluation indexes of models. We compare and analyze the F1score, accuracy of various models. As shown in Fig. 13, the F1score of CodnNet is slightly lower than that of DenseNet201, but higher than that of DenseNet121, DenseNet13, MobileNetV2, V3L, V3S models. As shown in Fig. 14, the accuracy of CodnNet is the same as that of DenseNet201 model, but higher than that of DenseNet121, DenseNet13, MobileNetV2, V3L, V3S. Considering the trade-off between algorithm complexity and classification accuracy, CodnNet has better comprehensive classification performance.

Table 6 shows the evaluation indexes of CodnNet model for 3-class classification after 5-fold cross-validation, with the average 97.9% precision, 97.4% recall, and 97.7% F1score, respectively. It can be seen that the CodnNet model has high recall and good generalization performance.

To show the comprehensive performance of each model for COVID-19 detection, we use the P–R curves to demonstrate the classification ability for class imbalance samples. As shown in Fig. 15, because the number of normal images in the dataset is greater than that of the COVID-19, and the number of the COVID-19 images is greater than that of the viral pneumonia, the P–R curve of normal class in each fold is the best, followed by COVID-19 and finally viral pneumonia, which is in line with the classification law, where the mean AP of the least 3 folds is greater than 99%, which shows that the CodnNet model has better classification performance for class imbalance. In addition, the proposed model analyzes 5-fold ROC curves for the samples of 3-class, and the ROC curve measures the predictive ability of positive samples, and the AUC value effectively reflects the probability of randomly selecting true positives over false positives.

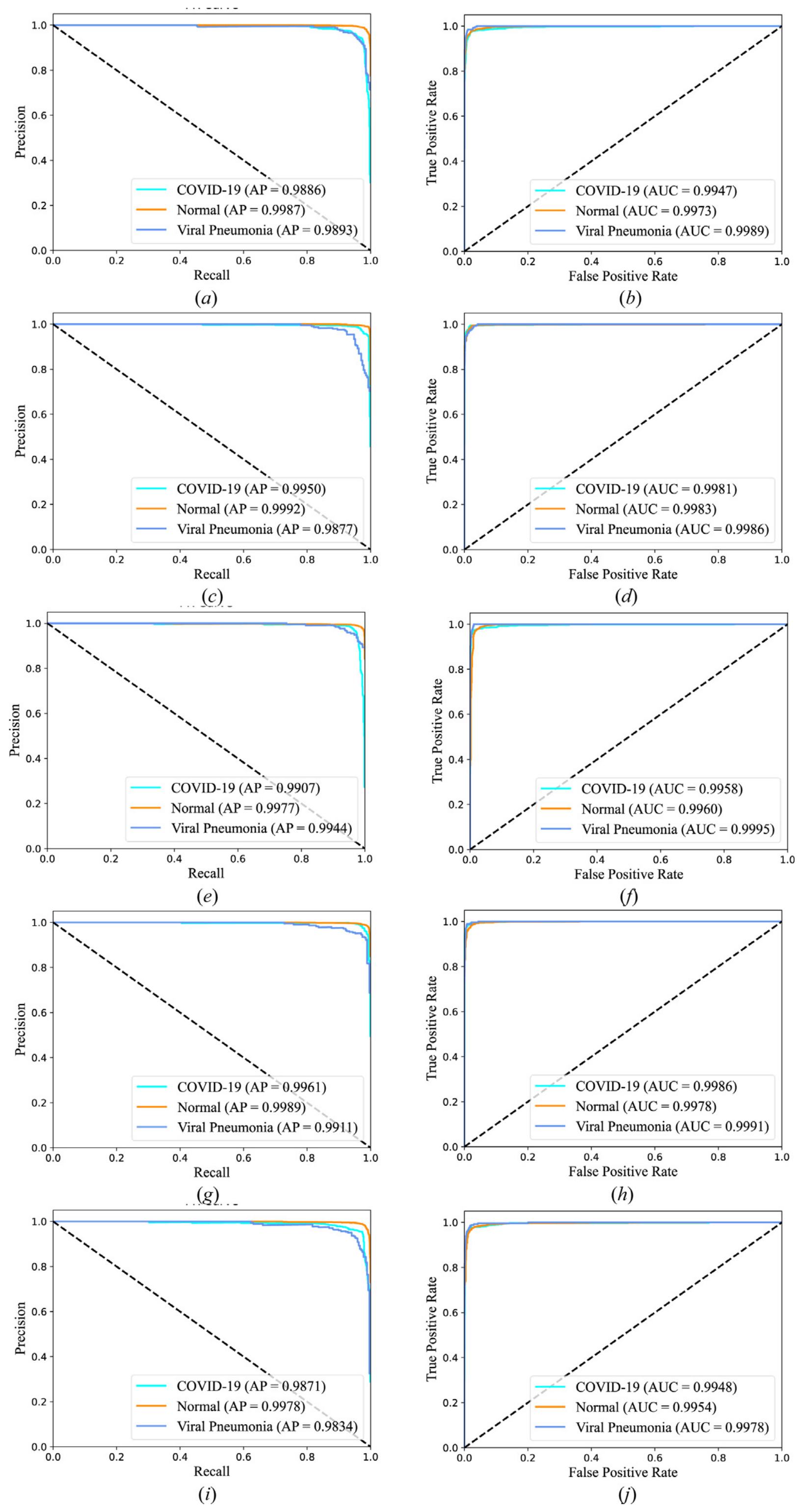

Fig. 15. Performance of CodnNet for 3-class CXR Images, where $(a)(c)(e)(g)(i)$ refers to P–R curve, and $(b)(d)(f)(h)(j)$ is ROC Curve Respectively.

The mean AUC value for the viral pneumonia class is the largest at 99.9%, followed by 99.7% for the normal and 99.6% for the COVID-19. Thus, the CodnNet model has a good rate of true positive COVID-19 detection.

Since COVID-19 is highly infectious, recall is highly important for COVID-19 detection. As shown in Table 7, the prediction results of different models on COVID-19 are given. In this paper, CodnNet achieves 97.6% recall on COVID-19, which is the best performance among all comparative models, and F1score reaches 97.8% and is the same as DenseNet201, which is better than other models. This shows that the CodnNet model has a higher recall on COVID-19, and has better classification accuracy and generalization performance compared with other models.

Table 7
Performance of the various models on each fold for COVID-19 detection.

Models	Precision	Recall	F1score	AP	AUC
DenseNet201	0.983	0.973	0.978	0.992	0.997
DenseNet121	0.980	0.973	0.977	0.991	0.997
DenseNet13	0.977	0.968	0.973	0.990	0.996
MobileNetV2	0.977	0.969	0.973	0.992	0.996
MobileNetV3L	0.982	0.971	0.977	0.994	0.997
MobileNetV3S	0.975	0.966	0.970	0.991	0.996
CodnNet	0.982	0.976	0.978	0.992	0.997

Table 8
The parameters and computational complexity of various models.

Model	Input resolution	MACs (G)	Params (M)
DenseNet201	256 × 256	4.37	18.10
DenseNet121	256 × 256	2.88	6.96
DenseNet13	256 × 256	0.33	0.19
MobileNetV2	256 × 256	0.32	2.23
MobileNetV3L	256 × 256	0.23	4.21
MobileNetV3S	256 × 256	0.06	1.52
CodnNet	256 × 256	0.06	0.26

5. Discussion

5.1. Computing complexity

Model complexity has the key impact on prediction error. Too low complexity leads to underfitting and too high complexity leads to overfitting. Model complexity includes time complexity and space complexity [44]. Time complexity, namely the number of model operations, can be measured by MAC (Memory Access Cost), which indicates the time required for model training or prediction. The spatial complexity includes the number of total model parameters. The more parameters of the model, the more memory space is required.

The comparison of algorithm complexity and number of parameters for each model is given in Table 8. The analysis shows that compared with DenseNet201, DenseNet121, DenseNet13, the MACs of CodnNet model are reduced by 7183%, 4700% and 450%. Compared with MobileNetV2, V3L, the MACs are reduced by 433% and 283%, It is the same as that of MobileNetV3S, but the number of parameters of CodnNet is reduced by 484% compared to MobileNetV3S. The main module of CodnNet model is the dense layers, and each dense layer increases the number of channels. Therefore, Codnblock layer reduces the number of output channels and width of the module via reduction of the number of convolutional layers in the dense layer, which in turn reduces the model complexity and parameters.

Too low model complexity also leads to degradation of classification performance. Therefore, model complexity and classification performance need to be considered as a whole. Fig. 16 shows the scatter plot of each model between accuracy and MACs, the CodnNet model is closest to the upper left position, indicating the best comprehensive performance. The comparison of different models consisting of accuracy and Params is shown in Fig. 17. Compared with the DenseNet201, DenseNet121, DenseNet13 models, the CodnNet model has the fewest number of model parameters, the highest classification accuracy, and CodnNet also has the highest classification accuracy and similar model complexity compared with MobileNetV2, V3L, and V3S. Therefore, CodnNet has a better trade-off between model complexity and classification performance.

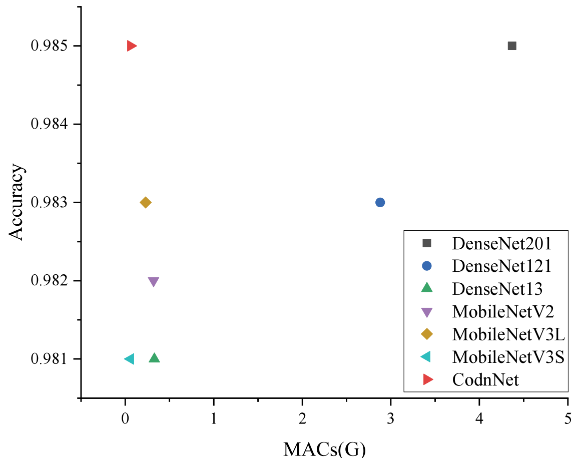

Fig. 16. Scatter plot of MACs and accuracy.

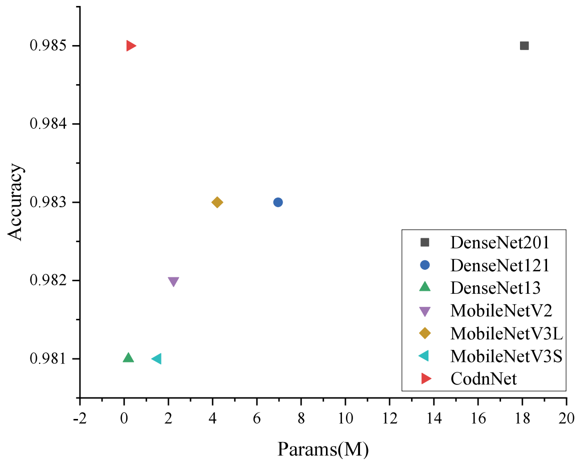

Fig. 17. Scatter plot of params and accuracy.

5.2. The attention features of various models for 3-class COVID-19 datasets

To visualize the infected region of COVID-19 accurately and rapidly, we use a visualization technique (Grad-CAM) to observe two-dimensional heatmap of lung CXR images. Grade-CAM presents "visual explanations" for decisions from a large class of CNN-based models [45]. The mapping of the gradient of the last convolutional layer to the original image can produce a coarse localization of important region in the image, which can effectively visualize the important attention features of image.

As shown in Fig. 18, the two-dimensional heatmap of the last layer of depthwise separable convolutional layer of CodnNet, the last layer of conventional convolutional layer of DenseNet13, DenseNet121 and DenseNet201 are given in the 3-class CXR images such as COVID-19, normal and viral pneumonia, respectively, and all the above models highly reuse the extracted features of each layer. For the COVID-19, CodnNet focuses more on the location of COVID-19 lesion features, while DenseNet focuses more on both sides of the lung features as the network depth increases. For the normal, CodnNet focuses on both sides of the lung features, which are also influenced by noises generated such as bones, and DenseNet focuses more on the intrapulmonary features as the network depth increases. For the viral pneumonia, CodnNet focuses on the focal features, while DenseNet gradually focuses on from all features of both sides of the lungs to the

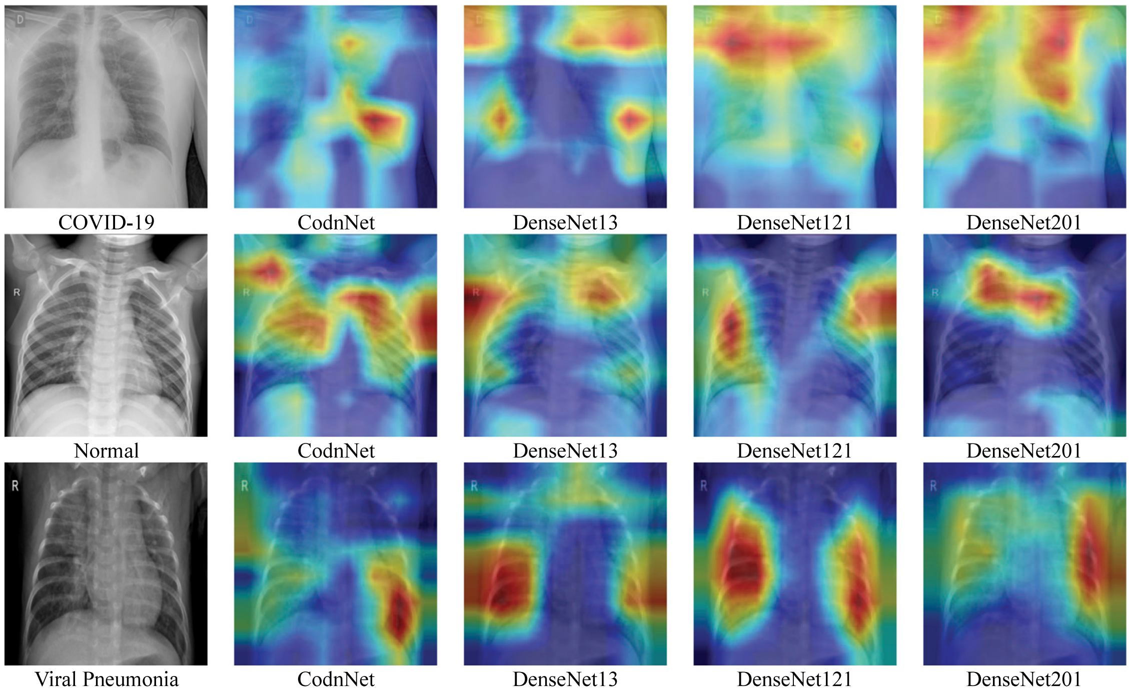

Fig. 18. The attention feature of various models for 3-class classification based on GRAD-CAM.

heavy focal features as the network depth increases. Therefore, it is clear that CodnNet can extract the focal features more quickly and accurately, and can obtain higher classification performance with lower complexity.

6. Conclusion

In this paper, we propose a lightweight CNN model, CodnNet, which uses depthwise separable convolution with large convolutional kernel to reduce model complexity, and increase the transform efficiency of in-depth or shallow features so as to enable more efficient reuse of feature maps for 3-class classification COVID-19 CXR images. Experimental results on the Kaggle COVID-19 dataset show that the average precision, recall, F1score and accuracy of CodnNet after 5-fold cross-validation are 97.9%, 97.4%, 97.7% and 98.5%, respectively. Compared with some typical lightweight CNN-based models, CodnNet has fewer model parameters and lower algorithm complexity, and the best generalization performance. Due to reduction of layers, CodnNet lacks global correlation and CXR image noise degrades the classification performance to some extent. Therefore, in the future, we will optimize the CodnNet model by adding a global attention mechanism to extract specific features, so that the model can pay more attention to the lung features of both sides and improve classification accuracy.

CRediT authorship contribution statement

Jingdong Yang: Supervision, Validation, Writing – reviewing & editing, Submission. **Lei Zhang:** Conceptualization, Methodology, Software, Data curation, Writing – original draft. **Xinjun Tang:** Supervision, Data curation. **Man Han:** Supervision, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This research is funded by the National Natural Science Foundation of China (81973749) and China Academy of Chinese Medical Sciences Innovation Fund (CI2021A01503)

Data availability

Data will be made available on request.

References

[1] A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90, http://dx.doi.org/10.1145/3065386, Available:.

[2] M.Z. Alom, T.M. Taha, C. Yakopcic, S. Westberg, V.K. Asari, The history began from alexnet: A comprehensive survey on deep learning approaches, 2018, arxiv:1803.01164.

[3] Z. Yang, Classification of picture art style based on VGGNET, J. Phys. Conf. Ser. 1774 (1) (2021) 012043, http://dx.doi.org/10.1088/1742-6596/1774/1/012043, Available:.

[4] J. Liang, Image classification based on RESNET, J. Phys. Conf. Ser. 1634 (1) (2020) 012110, http://dx.doi.org/10.1088/1742-6596/1634/1/012110, Available:.

[5] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 2261–2269, http://dx.doi.org/10.1109/CVPR.2017.243, Available:.

[6] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco An-dreetto, Hartwig Adam, MobileNets: Efficient con-volutional neural networks for mobile vision applications, 2017, arXiv:1704.04861.

[7] V.-T. Hoang, K.-H. Jo, Practical analysis on architecture of EfficientNet, in: 2021 14th International Conference on Human System Interaction, HSI, 2021, pp. 1–4, http://dx.doi.org/10.1109/HSI52170.2021.9538782, Available:.

[8] X. Luo, J. Zhang, Z. Li, R. Yang, Diagnosis of ulcerative colitis from endoscopic images based on deep learning, Biomed. Signal Process. Control 73 (2022) 103443, http://dx.doi.org/10.1016/j.bspc.2021.103443, Available:.

[9] C. Wang, W. Gong, J. Cheng, Y. Qian, DBLCNN: Dependency-based lightweight convolutional neural network for multi-classification of breast histopathology images, Biomed. Signal Process. Control 73 (2022) 103451, http://dx.doi.org/10.1016/j.bspc.2021.103451, Available:.

[10] C. Ouchicha, O. Ammor, M. Meknassi, CVDNet: A novel deep learning architecture for detection of coronavirus (Covid-19) from chest x-ray images, Chaos, Solitons Fractals 140 (2020) 110245, http://dx.doi.org/10.1016/j.chaos.2020.110245, Available:.

[11] M. Nour, Z. Cömert, K. Polat, A novel medical diagnosis model for COVID-19 infection detection based on deep features and Bayesian optimization, Appl. Soft Comput. 97 (2020) 106580, http://dx.doi.org/10.1016/j.asoc.2020.106580, Available:.

[12] N. Chowdhury, M. Rahman, M. Kabir, Pdcovidnet: a parallel-dilated convolutional neural network architecture for detecting COVID-19 from chest X-ray images, Health Inform. Sci. Syst. 8 (1) (2020) http://dx.doi.org/10.1007/s13755-020-00119-3, Available:.

[13] T. Mahmud, M. Rahman, S. Fattah, CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization, Comput. Biol. Med. 122 (2020) 103869, http://dx.doi.org/10.1016/j.compbiomed.2020.103869, Available:.

[14] M. Heidari, S. Mirniaharikandehei, A. Khuzani, G. Danala, Y. Qiu, B. Zheng, Improving the performance of CNN to predict the likelihood of COVID-19 using chest X-ray images with preprocessing algorithms, Int. J. Med. Inform. 144 (2020) 104284, http://dx.doi.org/10.1016/j.ijmedinf.2020.104284, Available:.

[15] H. Panwar, P. Gupta, M. Siddiqui, R. Morales-Menendez, P. Bhardwaj, V. Singh, A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-scan images, Chaos, Solitons Fractals 140 (2020) 110190, http://dx.doi.org/10.1016/j.chaos.2020.110190, Available:.

[16] R. Karthik, R. Menaka, H. M., Learning distinctive filters for COVID-19 detection from chest X-ray using shuffled residual CNN, Appl. Soft Comput. 99 (2021) 106744, http://dx.doi.org/10.1016/j.asoc.2020.106744, Available:.

[17] K. Shibly, S. Dey, M. Islam, M. Rahman, COVID faster R–CNN: A novel framework to diagnose novel coronavirus disease (COVID-19) in X-ray images, Inform. Med. Unlocked 20 (2020) 100405, http://dx.doi.org/10.1016/j.imu.2020.100405, Available:.

[18] V. Arora, E. Ng, R. Leekha, M. Darshan, A. Singh, Transfer learning-based approach for detecting COVID-19 ailment in lung CT scan, Comput. Biol. Med. 135 (2021) 104575, http://dx.doi.org/10.1016/j.compbiomed.2021.104575, Available:.

[19] L. Wang, Z. Lin, A. Wong, COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images, Sci. Rep. 10 (1) (2020) http://dx.doi.org/10.1038/s41598-020-76550-z, Available:.

[20] A. Khan, J. Shah, M. Bhat, CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images, Comput. Methods Programs Biomed. 196 (2020) 105581, http://dx.doi.org/10.1016/j.cmpb.2020.105581, Available:.

[21] G. Jia, H. Lam, Y. Xu, Classification of COVID-19 chest X-ray and CT images using a type of dynamic CNN modification method, Comput. Biol. Med. 134 (2021) 104425, http://dx.doi.org/10.1016/j.compbiomed.2021.104425, Available:.

[22] M.E.H. Chowdhury, et al., Can AI help in screening viral and COVID-19 pneumonia? IEEE Access 8 (2020) 132665–132676, http://dx.doi.org/10.1109/ACCESS.2020.3010287, Available:.

[23] T. Rahman, et al., Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images, Comput. Biol. Med. 132 (2021) 104319, http://dx.doi.org/10.1016/j.compbiomed.2021.104319, Available:.

[24] COVID-19 Radiography Database, Kaggle.com, 2022, [Online]. Available: https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database.

[25] COVID-19 DATABASE - SIRM, SIRM, 2022, [Online]. Available: https://www.sirm.org/category/senza-categoria/Covid-19/.

[26] Covid-chestxray-dataset: We are building an open database of COVID-19 cases with chest X-ray or CT images., GitHub, 2022, [Online]. Available: https://github.com/ieee8023/covid-chestxray-dataset.

[27] COVID-CXNet: COVID-CXNet: Diagnosing COVID-19 in frontal chest X-ray images using deep learning, GitHub, 2022, [Online]. Available: https://github.com/armiro/COVID-CxNet.

[28] B. Juba, H. Le, Precision-recall versus accuracy and the role of large data sets, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 4039–4048, http://dx.doi.org/10.1609/aaai.v33i01.33014039, Available:.

[29] J. Yang, M. Zhang, P. Liu, S. Yu, Multi-label rhinitis prediction using ensemble neural network chain with pre-training, Appl. Soft Comput. 122 (2022) 108839, http://dx.doi.org/10.1016/j.asoc.2022.108839, Available:.

[30] W. Mumtaz, A. Qayyum, A deep learning framework for automatic diagnosis of unipolar depression, Int. J. Med. Inform. 132 (2019) 103983, http://dx.doi.org/10.1016/j.ijmedinf.2019.103983, Available:.

[31] S. Pak, T. Oh, Application of receiver operating characteristic (ROC) curve for evaluation of diagnostic test performance, J. Veterinary Clin. 33 (2) (2016) 97–101, http://dx.doi.org/10.17555/jvc.2016.04.33.2.97, Available.

[32] J. Cook, V. Ramadas, When to consult precision-recall curves, SSRN Electron. J. (2019) http://dx.doi.org/10.2139/ssrn.3350582, Available:.

[33] Q. Zhou, L. Zhe, R. Brooke, M. Hudson, Y. Yuan, A relationship between the incremental values of area under the ROC curve and of area under the precision–recall curve, Diagnostic Progn. Res. 5 (1) (2021) http://dx.doi.org/10.1186/s41512-021-00102-w, Available:.

[34] S. Raschka, An overview of general performance metrics of binary classifier systems, 2014, arxiv:1410.5330.

[35] X. Liu, M. Pedersen, R. Wang, Survey of natural image enhancement techniques: Classification, evaluation, challenges, and perspectives, Digit. Signal Process. 127 (2022) 103547, http://dx.doi.org/10.1016/j.dsp.2022.103547, Available:.

[36] J. Glenn, YOLOv5, GitHub, 2022, [Online]. Available: https://github.com/ultralytics/yolov5.

[37] A. Howard, et al., Searching for MobileNetV3, in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV, 2019, pp. 1314–1324, http://dx.doi.org/10.1109/ICCV.2019.00140, Available:.

[38] A.F. Agarap, Deep learning using rectified linear units (relu), 2018, arxiv:1803.08375.

[39] Y. Kim, P. Panda, Revisiting batch normalization for training low-latency deep spiking neural networks from scratch, Front. Neurosci. 15 (2021) http://dx.doi.org/10.3389/fnins.2021.773954, Available:.

[40] M.P. Fodor, S.D. Bolboaca, L. Jantschi, Distribution of Molecules by Kinetic Energy Revisited, Bulletin of the University of Agricultural Sciences & Veterinary, 2013, http://dx.doi.org/10.15835/BUASVMCN-HORT:9176, Available:.

[41] S. Hahn, H. Choi, Understanding dropout as an optimization trick, Neurocomputing 398 (2020) 64–70, http://dx.doi.org/10.1016/j.neucom.2020.02.067, Available:.

[42] P. Refaeilzadeh, L. Tang, H. Liu, Cross-validation, Encyclop. Database Syst. (2016) 1–7, http://dx.doi.org/10.1007/978-1-4899-7993-3_565-2, Available:.

[43] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520, http://dx.doi.org/10.1109/CVPR.2018.00474, Available:.

[44] K. He, J. Sun, Convolutional neural networks at constrained time cost, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015, pp. 5353–5360, http://dx.doi.org/10.1109/CVPR.2015.7299173, Available:.

[45] R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, Int. J. Comput. Vis. 128 (2) (2019) 336–359, http://dx.doi.org/10.1007/s11263-019-01228-7, Available:.