Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
CrodenseNet: An efficient parallel cross DenseNet for COVID-19 infection detection

Jingdong Yang a,1,*, Lei Zhang a,1, Xinjun Tang b,*

a School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China
b Department of Respiratory and Critical Care Medicine, Zhongshan Hospital, Fudan University, Shanghai 200032, China

ABSTRACT

Purpose: At present, though the application of Convolution Neural Network (CNN) to detect COVID-19 infection significantly enhances detection performance and efficiency, it often causes low sensitivity and poor generalization performance. Methods: In this article, an effective CNN, CrodenseNet, is proposed for COVID-19 detection. CrodenseNet consists of two parallel DenseNet Blocks, each of which contains dilated convolutions with different expansion scales and traditional convolutions. We employ cross-dense connections and one-sided soft thresholding to the layers for filtering of noise-related features, and increase information interaction of local and global features. Results: Cross-validation experiments on COVID-19x dataset show that via CrodenseNet, the COVID-19 detection attains the precision of 96.7 ± 0.101, recall of 96.7 ± 0.010, F1-score of 97.3 ± 0.005, AP (area under P-R curve) of 0.991 ± 0.002, and AUC (area under ROC curve) of 0.996 ± 0.001. Conclusion: CrodenseNet outperforms a variety of state-of-the-art models in terms of evaluation metrics so it assists clinicians to prompt diagnosis of COVID-19 infection.

1. Introduction

Since December 2019, a novel Corona Virus Disease has been prevailing globally, and on February 11, 2020, the disease was officially named COVID-19 by the World Health Organization (WHO) [1]. Because of its rapid global spread, the WHO declared COVID-19 a global pandemic in March 2020 [2]. COVID-19 triggers neuroinflammation, hypoxia, vulnerability of the Blood Brain Barrier (BBB), increased predisposition to secondary infections and brain dysfunction [3]. Meanwhile, with the outbreak of COVID-19 across the globe, multiple researchers study COVID-19 via Artificial Intelligence (AI). Saleh Albahli et al. [4] used Natural Language Processing (NLP) techniques to perform a semantic analysis of three levels (negative, neutral, and positive) based on a Twitter dataset of 2 months to assess the feelings of Gulf countries towards the pandemic and the lockdown. The results showed that 50.5% of the countries had a neutral attitude towards the pandemic, 31.2% had a positive attitude and 18.3% had a negative attitude. Kumar et al. [5] proposed an AI-based automated image classification for sorting COVID-related medical waste streams from other waste. The result showed that the Support Vector Machine (SVM) performed best with 96.5% accuracy, and prevented the transmission of COVID-19 infection effectively. Alyasseri et al. [6] reviewed more than 200 papers published from December 2019 to April 2021 on the latest COVID-19 diagnosis using Deep Learning or Machine Learning and found that SVM was the most widely used machine learning for COVID-19 diagnosis or outbreak prediction, while CNN was the most widely used deep learning. Accuracy, sensitivity, and specificity were the most widely used measures in previous studies. Hasoon et al [7] applied three procedures to classify normal and COVID-19 images. The first procedure was preprocessing, including noise removal, thresholding, and morphological operation. The second was Region of Interest (ROI) detection and segmentation, in which Local Binary Pattern (LBP) and Histogram of Gradient (HOG) were employed to extract edge feature and Haralick was used for extraction of texture feature. Finally, K-Nearest Neighbors (KNN) and SVM were used for classification. The results showed LBP-KNN model with an average accuracy of 98.66% outperformed the other models. It showed that machine learning had achieved higher performance in COVID-19 detection, but unsatisfactory robustness due to feature engineering. With the development of deep learning, CNN is usually used instead of feature engineering to accomplish end-to-end automatic detection.

The main contribution in this paper is as follows.
To extract COVID-19 features with different dimensions and scales, a parallel cross DenseNet, composed of dilated convolutions and traditional convolutions is proposed to enlarge receptive fields for extraction of more features in global and semantic levels. The one-sided soft thresholding is employed to remove noise-related features, and auxiliary classification and regularization is integrated to reduce overfitting. Cross-validation experiments on COVID-19x chest X-ray images datasets show that the proposed model outperforms the state-of-the-art CNN models in classification accuracy and generalization performance.

Grad-CAM visualization technology is applied to display the key areas of feature extraction, and aid the clinicians in clinical diagnosis and analysis.

In this paper, Section II discusses related works. Section III introduces the overall framework and implementation of the proposed model. Section IV presents experimental setup including dataset description and hyperparameter Setting. Section V is experimental results and analysis. Section VI describes the discussion on computational complexity and features of visualization. We conclude the main traits and shortcomings of proposed model in Section VII.

2. Related works

CNN have been used in multiple researches to implement classification on COVID-19 as shown in Table 1. Al-Waisy et al. [8] used Contrast-Limited Adaptive Histogram Equalization (CLAHE) and Butterworth bandpass filter to enhance the contrast and eliminate the noise in CX-R images, respectively. Then, input to two different deep learning methods (a deep belief network and a convolutional deep belief network) to distinguish healthy and COVID-19 infected individuals. The experimental results showed this way can diagnose patients with COVID-19 with a detection accuracy rate of 99.93% in the large-scale dataset. Barshooi et al. [9] first expanded the dataset by combining the traditional data augmentation techniques with the Generative Adversarial Networks (GANs) and then applied different filters (Sobel, Laplacian of Gaussian (LoG) and the Gabor filters) for feature extraction, using the output of the filters as well as the original data are considered as new data and input into 6 migration models for training (AlexNet, GoogleNet, VGG-19, ShuffleNet V2, DenseNet-121 and DenseNet-201). The experiment shows that the combination of Gabor and DenseNet-201 best, with 2-class classification accuracy of 98.5%. The methods use specific filter to reduce the noises of image on specified datasets amid data preprocessing to improve classification performance. However, the variation of distributions and types of the dataset will bear on the classification results to a certain extent.

### Table 1

| Researchers          | Models                          | Overall Accuracy | F1-score of COVID-19 |
|----------------------|---------------------------------|------------------|-----------------------|
| A. S. Al-Waisy et al [8] | 12000,12000 DBN + CDBN          | 0.993            | 0.999                 |
| A. Barshooi et al. [9] | 360,4200 DenseNet-201           | 0.985            | 0.978                 |
| S. Rajpal et al. [10] | 520,520,520 ResNet50            | 0.974 ± 0.02     | 0.987                 |
| S. Karakanis et al. [11] | 275,270,275 CNN                | 0.983            | /                     |
| T. Ozturk et al. [12] | 125,500,500 DarkCovidNet        | 0.8702           | 0.8737                |
| C. Ouchicha et al. [13] | 219,1341,1345 CVDNet           | 0.9669           | 0.9664                |
| I. Wang et al. [14]    | 266,8086,5538 COVID-Net         | 0.933            | 0.9268                |
| P. Kedia et al. [15]   | 266,8086,5538 CoVNet-19         | 0.9828           | 0.99                   |

Rajpal et al. [10] created a segmented feature extraction and classification method for detection of COVID-19 chest X-ray images using three modules. The first module extracted 2048 feature vectors using ResNet-50. The second module extracted 252 frequency-domain and texture features, furtherly reduced to 16. The third module connected the features of the first and second modules and classified using fully connected layer. Using this method, the performance improved. However, when the data distribution changed, the frequency-domain or texture features changed as well, and the classification performance was degraded. Kedia et al. [11] created an integrated structural model (CoVNet-19) that combined two depth models (e.g. VGG-19 and DenseNet-212) to extract in-depth features from chest X-ray images. The SVM classifier was applied for binary classification and 3-class classification, yielding accuracy of 0.9971 and 0.9828, respectively. By this method a deep learning model is used to extract in-depth features of COVID-19 image, and a machine learning model is used for classification. Because the feature extractor and feature classifier are heterogeneous, classification performance may decrease in case of different distributions of samples.

Karakanis et al. [12] first used GAN to expand datasets, and then applied ResNet-8 model to build two lightweight deep learning networks for binary classification and 3-class classification, respectively. The binary classification yielded an accuracy of 0.987, and 3-class classification accuracy of 0.983. Because the data generated by GAN and the original data fell into the same distribution, it was difficult to generate effective data when a large amount of data had different distributions. Therefore, we should choose a large dataset with multiple data sets fused for training and testing, which can show the robustness of the model well. Ozturk et al. [13] improved Darknet-19 model by removing layers and adding filters for binary classification and multiple classification, yielding accuracy of 0.9808 and 0.8702, respectively. Darknet-19 is a YOLOV3-based feature extraction network that can effectively extract local pixel position features, but it is not sensitive to global features. Both ResNet-8 and Darknet-19 are lightweight models with fewer parameters, but they may have a decreased classification performance for hard samples. Therefore, the trade-off between the number of parameters and classification performance should be taken into account.

Ouchicha et al. [14] proposed a 3-class image classification model that connected the residual blocks of two convolution kernels with different scales in parallel and added cross residual links to reuse the features of different scales. The model extracted features using two larger convolutional kernels, which increased complexity of model. The residual network excels in facilitating the back propagation of gradient, but it is unable to make full use of the different sizes of extracted features, causing a decrease of classification performance. Wang et al. [15] applied reinforcement learning to generate a special model for COVID-19 datasets, and used the channel-wise convolutions to replace traditional convolution to reduce complexity of the model. Such models show good performance on some specific dataset, but the increase of numbers of dataset would affect the overall performance of classification.

In summary, the research on medical image has some limitations. For example, the convolutional kernel of fixed size is unable to extract the features of different sizes. The feature map of various dimensions is unable to have impact on classification fully, and reduce the efficiency in feature utilization. Meanwhile, those above models have deficiency on an effective filtering mechanism resulting in the reduction of feature transmission ratio. Therefore, we propose an end-to-end model whose feature extractor and classifier are trained at the same time to maximize complex interplay between them. We also use a parallel cross DenseNet, which highly reuses the extracted features and achieves better performance. We employ a one-sided soft thresholding structure to automatically filter features further to reduce noises for different feature layers.
3. A parallel DenseNet model with Cross-Dense connections

3.1. Architecture of CrodenseNet model

We propose a novel model CrodenseNet based on DenseNet [16] that applies dilated convolution layers [17] and traditional convolution layers to extract COVID-19 features from receptive fields with different scales. There are a total of 3 major innovations in the model structure: cross-dense connection, one-sided soft threshold and auxiliary classifier.

Cross-dense connections

As is shown in Fig. 1, on the basis of DenseNet, we build a novel Block that has two DenseBlocks in parallel, then add a cross-dense connections between the two DenseBlock channels, and finally we use conventional convolution with a convolution kernel size of $3 \times 3$ in one side channel and dilated convolution with a convolution kernel size greater than $3 \times 3$ in the other side channel.

As a result, the convolution layer in each DenseBlock can accept the output feature map of the current channel in parallel with the output of channel. This block can extract deep and shallow features with different scales via complex cross-dense connections. The output of each layer can be expressed as follows.

$$x^a_l = \text{ReLU}(\text{BN}(\text{Conv}(\left[x^a_0, x^a_2, \ldots, x^a_{l-1}, x^b_0, x^b_1, \ldots, x^b_l\right]))) \quad (1)$$

where $x^a$ or $x^b$ dentoes the $a$th or $b$th channel, $[x^a_0, x^a_2, \ldots, x^a_{l-1}, x^b_0, x^b_1, \ldots, x^b_l]$ is the output concatenation from the 0th layer to ($l-1$)th layer. BN is BatchNormalization layer [18], ReLU is Linear rectification function [19] and Conv is convolutional layer.

The Bottleneck layer of each channel in original DenseBlock is the $1 \times 1$ convolution layer combined with the growth rate. First, the $1 \times 1$ convolution layer is used to generate the feature graphs with channels of $n \times$ growth rate. Then the $N \times N$ convolution layer is used to generate the feature graphs with channel of growth rate, which reduces the number of feature graphs and model parameters. Because of the addition of cross dense connections, the $l$th layer of Block has 2 $l-1$ inputs and Block of $l$ layer has all $2^l$ connections. It consists of splicing of output feature graphs in first two convolutional layers of DenseBlock. If the number of channels from input is $x$ and the growth rate is $a$, the final number of output channels will be $2x + (2^l - 1)a$.

One-sided soft thresholding

As shown in Fig. 2, we apply one-sided soft thresholding, a kind of nonlinear transformation with channel attention-based mechanism, to cross dense connections. The pixels that is smaller than $\tau$ are set as zero, while the pixels larger than $\tau$, are set as minus $\tau$. The improved architecture of attention-based mechanism is shown in Fig. 3.

According to SENet [20], So, we apply the Global Average Pooling layer (GAP) to compress the feature map of each channel, yielding a series of real numbers with the same channel number. Then, by using a fully connected layer and ReLU transformation, the number of channels is reduced by 8 times for reduction of model complexity. We use the fully connection layer to restore the number of channels to the original number. And the softmax transformation is applied to each real number in the range (0,1), and the sum is 1. Then, the $\tau$ of each channel is calculated by multiplying the real number after the GAP by the channel attention-based weight. The purpose of softmax transformation is to map each $\tau$ in the range of its each channel value. According to DRSN-CS [21], we filter the original images for denoising using soft-thresholding, and combine with ReLU non-linear transformation to build a one-side soft-thresholding structure, and activate the feature maps on each channel via a nonlinear transformation with one-sided soft thresholding.

![Fig. 1. The architecture of the cross-dense connections.](image1)

![Fig. 2. Illustration of one-sided soft thresholding.](image2)

![Fig. 3. Attention Mechanism.](image3)
concerning \( r \). Therefore, the features greater than their respective \( r \) in different channels will be set as the current feature minus \( r \) and those smaller than \( r \) are set as zero.

The method can learn the importance of each channel automatically, eliminate interference pixels, and reduce the influence on the subsequent convolutional layers.

**Crodense Block**

We name the combination of cross dense connections and nonlinear transformation with one-sided soft thresholding as CrodenseBlock in this study.

**Lemma:** Assumes that for inputs \( x^i, x^h, x \in \mathbb{R}^{i \times h \times w} \), one-sided soft thresholding is \( r^i, r^h \), and convolution kernels are \( K^i, K^h \) respectively. Convolution kernels are \( K^i, K^h \), \( r \) denoting a convolutional operation. Then the output \( o \) of one-sided channel after CrodenseBlock of one layer is as follows:

\[
-o^\tau(h, w) = \left[ x^\tau, \max\{x^i - r^i, 0\} \right] \text{max}\{(x^h K^h)(h, w), 0\}
\]

As shown in Fig. 3, GAP in Attention Mechanism module represents the global average pooling layer, with \( w \) being the fully connected layer parameters, \( r \) representing the features after the one-sided soft thresholding, and \( y \) representing the result after the convolution operation. Then \( \tau \) is as follows:

\[
\tau = \text{Softmax}(\omega^T \text{GAP}(x)) \times \text{GAP}(x)
\]

Transform the feature map via one-sided soft thresholding.

\[
r = \max\{x - r, 0\}
\]

Conduct convolution operation and nonlinear transformation ReLU on feature maps.

\[
y(h, w) = \max\{(x K)(h, w), 0\}
\]

Concenate the feature maps of two channels by cross dense connections.

\[
o^\tau(h, w) = \left[ x^\tau, r^i, r^h \right]
\]

When \( x = [\theta, \mu] \), where \( \theta > \tau, \mu < \tau \).

\[
r = [0 - \tau, 0]
\]

\[
o^\tau(h, w) = \left[ \theta^i, \mu^i, \theta^h - \tau^h, 0, y^\tau(h, w) \right]
\]

As illustrated in Fig. 4, CrodenseBlock1 module is made up of two parallel channels, with a total of 6 layers and each layer consisting of two parallel convolutional layers. It has a total of 72 connections because it has 6 layers (n). The growth rate (g) has been set to 16. The traditional convolution layer is used in the left channel, with a convolution kernel size of \( 3 \times 3 \), which uses a small receptive field to extract small-scale features. In the right channel, the traditional convolution layer is replaced by a dilated convolution layer, and the expansion index (d) is 8, with seven zeros filled between the weights of each convolution kernel, resulting in convolution kernel size of \( 17 \times 17 \). This method makes use of the large receptive field to extract large-scale features. The feature maps of each convolution layer are transformed by a nonlinear transformation with one-sided soft thresholding and transmitted to the subsequent convolution layer via the cross-dense connections. Different in-depth features are continuously added during feature transmission, and different feature maps can be highly reused in two channels. Therefore, the model integrates large-scale context information to improve classification accuracy of the model.

As shown in Fig. 5, CrodenseNet applies four different CrodenseBlock blocks to extract features from feature maps with various sizes to the greatest extent possible. Following each CrodenseBlock, the feature outputs of the left and right channels are connected to different transition layers. The left transition layer consists of a \( 1 \times 1 \) convolution layer and a \( 2 \times 2 \) average pooling layer, while the right transition layer consists of a \( 1 \times 1 \) convolution layer and a \( 3 \times 3 \) convolutional layer with a stride (s) of 2. On the other side of channel, the convolution layer with a stride of 2 is used to replace average pooling layer to prevent the loss of some spatial characteristics when the average pooling layer reduces the feature scale.

**Auxiliary Classifier**

Following four CrodenseBlock structures, we add two identical auxiliary classifiers in the outputs of two channels to increase more effective gradient signals and an additional regularization to curb overfitting in terms of GoogleNet [22]. The feature maps of two channels are concatenated and fed into the main classifier for final decisions. To prevent overfitting, all classifiers have three fully connected layers, with a Dropout layer [23] added after the first two fully connected layers. Following the final fully connected layer, the feature map is fed into the softmax [24] function, which yields the probability distribution for the corresponding class.

### 3.2. Implementation of CrodenseNet model

The CrodenseNet model in this paper uses three cross-entropy loss functions to calculate the loss between the SoftMax output probability and the actual probability. As a result, CrodenseNet includes two auxiliary classifiers (Classifier1 and Classifier2) with loss of \( L_1 \) and \( L_2 \), as well as a primary classifier (Classifier3) with loss of \( L_3 \). The auxiliary classifier is used to improve effectiveness in training stage. The different weights are assigned to the three cross-entropy loss functions, with two auxiliary classifiers of 0.1, the main classifier of 0.8. Then, in reverse transmission, the formula of loss is as follows:

\[
L = \frac{1}{n} \sum_{i=1}^{n} \left( - \sum_{j=1}^{k} y_i \log p_j \right)
\]

where \( n \) is the number of images, \( k \) is the number of classes, \( y \) is the image label, and \( p \) is the predicted probability.

\[
L_{all} = 0.1 \times L_1 + 0.1 \times L_2 + 0.8 \times L_3
\]

The optimizer employs stochastic gradient descent (SGD), with a learning rate of 1e-3(\( \mu \)). The training epoch is 100 epoch(\( e \)), and the batch(\( \beta \)) is 32. In each epoch, \( \beta \) cases of chest X-ray images are selected for training, and after training \( t \) times (each chest X-ray images in the training set was trained once), the entire validation set is tested, and the test result (\( J \)) is calculated based on the output of main classifier. As the test result, the F1-score is chosen as evaluation metrics on COVID-19. When \( \lambda \) is reaching the highest value, save the current training
parameters ($\omega$) and enter the next epoch training until training over. Load $\omega$ for testing and evaluate classification performance. Table 2 depicts the detailed process.

4. Experimental setup

4.1. Dataset description

The datasets in this paper are from the COVID-19 X datasets, which includes 3 classes e.g. COVID-19, normal, and viral pneumonia. They are made up of five different public COVID-19 Datasets: (1) COVID-19 Image Data Collection [25], (2) Fig. 1 COVID-19 Chest X-ray images Dataset [26], (3) ActualMed COVID-19 Chest X-ray images Dataset [27], (4) RSNA Pneumonia Detection Challenge dataset [28], and (5) COVID-19 radiography database [29]. There is a total collection of 19,164 Chest X-ray images, which include 4187 COVID-19 images, 8851 normal images, and 6126 viral pneumonia images. Table 3 shows the detailed distribution of each dataset. The chest X-ray images of COVIDx are shown in Fig. 6.

As shown in Fig. 7, we randomly selected 100 images from each class, 300 images in total, and explored t-distributed stochastic neighbor embedding (t-SNE) to downscale all samples to 2 dimensions. As can be seen, most of the images of different classes are discriminative, with COVID-19 on the left, normal on the lower right, and viral pneumonia on the upper right. However, since this dataset comes from 5 small datasets with different distributions, there are maybe overlap of the data distribution for 3-class images classes.

4.2. Evaluation metrics and data augmentation

Precision, Sensitivity, F1-score, Accuracy, AP of area under P-R curve, and AUC of area under ROC curve are considered as evaluation metrics in this paper. TP, TN, FP, and FN denote true positive, true negative, false positive, and false negative, respectively. The details are as follows.

$$\text{Precision} = \frac{TP}{TP + FP}$$  \hspace{1cm} (11)

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$  \hspace{1cm} (12)

$$F1 - \text{score} = 2 \times \frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}}$$  \hspace{1cm} (13)

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + TP + FP}$$  \hspace{1cm} (14)

The Chest X-ray images in this paper are from different COVID-19 datasets. As a result, we first reduce the resolution of Chest X-ray images to 260 × 260, then horizontally rotate in the interval of (-10°, +10°) and flip randomly, and finally crop image in center to obtain the remaining 256 × 256 pixels. The augmentation effectively removes text interference and highlight ROI. Before being fed into the model, all the

Table 2

| Input Chest X-Ray images Training set $\delta_1$, Valid set $\delta_2$ |
|---------------------------------------------------------------------|
| $\mu$ → the CNN initial learning rate                                 |
| $\xi$ → iteration steps in the CNN                                    |
| $\epsilon$ → the CNN maximum number of iterations                     |
| $\alpha$ → iteration steps in a training                              |
| $\beta$ → the number of images covered in a training                  |
| $\eta_1$ → the maximum number of iterations in a training             |
| $\eta_2$ → the maximum number of iterations in a training             |
| $\lambda$ → variables of saving accuracy ($\lambda \leftarrow 0$)      |

| Output $\omega$ → CNN weight                                           |

Start
1. Initialize the CNN parameters: $\mu$, $\xi$, $\beta$, $\lambda$
2. Light preprocessing the images with resolution of $3 \times 256 \times 256$
3. Train the CNN and compute 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 $\alpha = 1$ 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. if the F1 score of COVID-19 > $\lambda$ do
15. Save $\omega$ and $\lambda \leftarrow$ F1 score of COVID-19
16. end
End

Table 3

| ID | COVID-19 | NORMAL | PNEUMONIA | TOTAL IMAGES |
|----|----------|--------|-----------|--------------|
| 1  | 478      | N/A    | 57        | 535          |
| 2  | 35       | N/A    | 57        | 92           |
| 3  | 58       | N/A    | N/A       | 58           |
| 4  | N/A      | 8851   | 6012      | 14863        |
| 5  | 3616     | N/A    | N/A       | 3616         |
chopped images need normalization, and the RGB mean and standard deviation are set to 0.485, 0.456, 0.406, and 0.229, 0.224, 0.225, respectively.

4.3. Hyperparameter setting

The detailed hyperparameters of CrodenseNet are shown in Table 4. The size of the input images is $3 \times 256 \times 256$. The process of feature extraction is given as follows. The left channel goes through the convolution layer with a convolution kernel of $3 \times 3$ and a stride ($s$) of 2; the right channel goes through the dilated convolution layer with a scale ($d$) of 16 and a stride ($s$) of 2, so the convolution kernel size becomes $33 \times 33$. The feature map has grown to $64 \times 128 \times 128$. A maximum pooling layer of $2 \times 2$ is linked on the left, and a convolution layer of $3 \times 3$ with a stride of 2 is linked on the right. Following this step, the feature map is reduced to $64 \times 64 \times 64$. The images are then processed by the first CrodenseBlock layer, which has a growth rate ($g$) of 16, the number of layers ($n$) of 6, and an expansion rate ($d$) of 8 for the right channel. The feature maps are changed to $304 \times 64 \times 64$. The left and right feature maps each passing through a convolution layer of $1 \times 1$, reduce their number of channels to $152 \times 64 \times 64$. The left side is then passed through a $2 \times 2$ average pooling layer, and the right side is passed through a $3 \times 3$ convolution layer with a stride of 2, resulting in a feature map of $152 \times 32 \times 32$. Then through the second CrodenseBlock layer, with a growth rate of 16 and the number of layers of 12, the expansion rate of the right channel is 2. The feature map becomes $672 \times 32 \times 32$. Again, the feature map is $336 \times 16 \times 16$ after passing through the second transition layer.

The feature map is then $1424 \times 16 \times 16$ after passing through the third CrodenseBlock layer, whose growth rate is 16, the number of layers is 24, and the expansion rate of right channel is 2. The feature map becomes $712 \times 8 \times 8$ after the third transition layer. Through the fourth CrodenseBlock layer, whose growth rate is 16, the number of layers is 16, the convolution layer of right channel is changed to $3 \times 3$ convolution layer, and the feature map becomes $1920 \times 8 \times 8$. Then, using convolution layer of $1 \times 1$, the number of channels is reduced to $960 \times 8 \times 8$. The structure of the left and right auxiliary classifiers is as follows. First, the feature map is globally pooled to $960 \times 1 \times 1$; after that, it passes through the first fully connected layer and Dropout layer, and the feature map becomes $512 \times 1 \times 1$. The feature map is reduced to $256 \times 1 \times 1$ after passing through the second fully connected layer and Dropout layer. Finally, it becomes $3 \times 1 \times 1$ after passing through the third fully connected layer and Softmax layer.

The main classifier is structured as follows: the feature maps of left and right channels are concatenated into $1920 \times 8 \times 8$. The feature map changes to $1024 \times 4 \times 4$ after a $3 \times 3$ convolution layer with a stride of 2, $512 \times 4 \times 4$ after using a $3 \times 3$ convolution layer with a stride of 1. The main classifier reduces number of channels for fusing feature in formation of each channel. The feature map is reduced to $256 \times 1 \times 1$ after global average pooling. It becomes $512 \times 1 \times 1$ after the first fully connected and Dropout layer, and $256 \times 1 \times 1$ after the second fully connected and Dropout layer; finally, the feature map becomes $3 \times 1 \times 1$ after the third fully connected layer and Softmax layer.
Fig. 8 shows the confusion matrix of 5-fold cross-validation, where each fold consists of 832 cases of COVID-19 images. CroDenseNet has the highest average recall of 0.967 on COVID-19, which can help clinical physicians with better detection rates and timely decision-making. Compared to DenseNet, CvdNet, CovidNet, and DarkCovidNet, CroDenseNet has a high positive detection rate on COVID-19 and is suitable for COVID-19 rapid testing. The AP of each fold on COVID-19 is also the highest (0.991), indicating that CroDenseNet achieves good classification performance when the number of COVID-19 instances is significantly lower than that of normal or viral pneumonia class.

Fig. 9 shows that evaluation metrics of CroDenseNet, i.e., Precision, Recall, and F1-score, on COVID-19 detection are the highest for 5-fold cross-validation. Table 5 provides detailed statistics for classification performance of various models on COVID-19 detection. Compared to DenseNet, CvdNet, CovidNet, and DarkCovidNet, CroDenseNet has improved Precision by 0.7%, 2.3%, 5%, 1.8%, respectively, F1-score by 1.1%, 2.1%, 5.8%, 1.9% to 0.973%, Recall by 1.6%, 1.9%, 6.5%, 1.3%, AUC by 0.1%, 0.1%, 0.9%, 0.1% to 0.996%, and AP by 1%, 0.4%, 2.9%, 0.5%, respectively. Generalization performance of CroDenseNet outperforms the four state-of-the-art models, and achieves average recall of 0.967 on COVID-19, which can help clinical physicians detect potential false negative COVID-19 patients effectively and timely, and reduces COVID-19 transmission risks via early diagnosis.

Table 4

| Configuration of CroDenseNet hyperparameters. |
|-----------------------------------------------|
| **Name** | **Layers** | **size of filters** | **Channel1** | **Channel2** | **Output shape** |
| Feature | Input | – | Input | – | \(3 \times 256 \times 256\) |
| Conv | 3 \(\times 3, s = 2, p = 2\) | – | Conv | 3 \(\times 3, s = 2, p = 16, d = 16\) | \(64 \times 128 \times 128\) |
| MP | 2 \(\times 2, s = 2\) | – | Conv | 3 \(\times 3, s = 2, p = 1\) | \(64 \times 64 \times 64\) |
| CroDenseBlock1 | g = 16, n = 6, d = 8 | – | Conv | 1 \(\times 1, s = 1\) | \(304 \times 64 \times 64\) |
| AP | 1 \(\times 1, s = 1\) | \(3 \times 3\) | Conv | 3 \(\times 3, s = 2, p = 1\) | \(152 \times 64 \times 64\) |
| CroDenseBlock12 | g = 16, n = 12, d = 2 | – | Conv | 1 \(\times 1, s = 1\) | \(336 \times 32 \times 32\) |
| Conv | 2 \(\times 2, s = 2\) | \(3 \times 3\) | Conv | 1 \(\times 1, s = 1\) | \(336 \times 16 \times 16\) |
| CroDenseBlock13 | g = 16, n = 24, d = 2 | – | Conv | 1 \(\times 1, s = 1\) | \(1424 \times 16 \times 16\) |
| Conv | 2 \(\times 2, s = 2\) | \(3 \times 3\) | Conv | 1 \(\times 1, s = 1\) | \(712 \times 16 \times 16\) |
| CroDenseBlock14 | g = 16, n = 16, d = 1 | – | Conv | 1 \(\times 1, s = 1\) | \(712 \times 8 \times 8\) |
| Classifier1 | GAP | – | GAP | – | \(960 \times 1 \times 1\) |
| FC | – | – | FC | – | \(512 \times 1 \times 1\) |
| Dropout | p = 0.4 | – | Dropout | p = 0.4 | \(256 \times 1 \times 1\) |
| FC | – | – | FC | – | \(3 \times 1 \times 1\) |
| Softmax | – | – | Softmax | – | \(3 \times 1 \times 1\) |
| Classifier2 | Concat | – | Concat | – | \(1920 \times 8 \times 8\) |
| Conv | 3 \(\times 3, s = 2, p = 1\) | – | Conv | 3 \(\times 3, s = 1, p = 1\) | \(1024 \times 4 \times 4\) |
| GAP | – | – | GAP | – | \(512 \times 4 \times 4\) |
| FC | – | – | FC | – | \(512 \times 1 \times 1\) |
| Dropout | p = 0.4 | – | Dropout | p = 0.4 | \(256 \times 1 \times 1\) |
| FC | – | – | FC | – | \(3 \times 1 \times 1\) |
| Softmax | – | – | Softmax | – | \(3 \times 1 \times 1\) |

5. Experimental results and analysis

5.1. Performance of COVID-19 detection

This paper conducts a 3-class classification experiment on the COVIDx datasets, with 70% as the training data, 10% as validation, and 20% as test. Fig. 8 shows the confusion matrix of 5-fold cross-validation, as well as PR and ROC curves for detection of COVID-19 infection. Each fold consists of 832 cases of COVID-19 images. CroDenseNet has the highest AUC (0.996) of each fold on COVID-19 when compared to DenseNet, CvdNet, CovidNet, and DarkCovidNet, indicating that CroDenseNet has a high positive detection rate on COVID-19 and is suitable for COVID-19 rapid testing. The AP of each fold on COVID-19 is also the highest (0.991), indicating that CroDenseNet achieves good classification performance when the number of COVID-19 instances is significantly lower than that of normal or viral pneumonia class.

5.2. Performance of CroDenseNet model

Fig. 10 shows 5-fold cross-validation results on 3-class classification. Because CroDenseNet focuses on prediction of COVID-19 infection, performance in prediction of normal and viral pneumonia has a downside, lowering the overall performance. However, the average of Precision, Recall, F1-Score, AP, and AUC are higher than those of other state-of-the-art models. Table 6 shows various evaluation metrics of various models. Compared to DenseNet121, CvdNet, CovidNet, and DarkCovidNet, CroDenseNet has improved the precision by 0.5%, 1.5%, 6.7%, and 2.3% respectively, recall by 1.1%, 2.3%, 7.1%, and 1.9% to 0.935%, F1-score by 0.9%, 2%, 6.8%, and 2.1% to 0.937%, accuracy by 0.8%, 1.9%, 6.9%, and 2.1%, respectively. Therefore, generalization performance of CroDenseNet outperforms the four state-of-the-art models on normal and viral pneumonia detection in terms of evaluation metrics.

CroDenseNet model has a total of 43,571,503 parameters, the majority of which are applied for the nonlinear transformation with one-sided soft thresholding and fully connected layers of final classifier. As we use a large number of 1 \(\times 1\) convolution layers, the time complexity is kept to a minimum. As a result, CroDenseNet can perform quick prediction of COVID-19 infection on batch Chest X-ray images.

5.3. Ablation study

To analyze performance of each block of CroDenseNet, we improve the basic parallel densenet via addition of different blocks, such as cross-dense connection (CD), Dilated convolution (DC), one-sided soft thresholding (ST), and multiple classifier (MC) blocks for ablation study.
Fig. 8. 5-fold Confusion Matrix of CrodenseNet and PR and ROC curve for COVID-19 detection.
The model complexity is the key factor influencing prediction error. Low complexity leads to underfitting, while high complexity causes overfitting. Time complexity and model parameters can often be used to represent model complexity [30]. The time complexity, or number of model operations, is measured in floating-point operations (Flops) and represents the time required to train or predict. Model parameters are the derivatives of convolution layer and fully connected layer during model training. The greater the number of model parameters, the greater the amount of memory required. The time complexity and model parameters are calculated as follows.

6. Discussion

6.1. Computational complexity

The model complexity is the key factor influencing prediction error. Low complexity leads to underfitting, while high complexity causes overfitting. Time complexity and model parameters can often be used to represent model complexity [30].
\[ O \left( \sum_{l=1}^{D} M^2 K_l^2 C_{l-1} C_l \right) \]  
\[ \sum_{l=1}^{D} K_l^2 C_{l-1} C_l \]  
where M is the length of feature maps, K is the length of convolutional kernel, \( l = [1, \ldots, D] \) is the number of convolutional layers, and \( C_l \) represents the number of output channels of the lth convolutional layer.

As shown in Table 6, the limitation of the CrodenseNet is the use of a large number of fully connected layers, which results in an increase in model parameters and time complexity. Hence, we plot a radar chart for each model to showcase the trade-off between time complexity and qualitative performance as shown in Fig. 13. It can be seen that our proposed model CrodenseNet has almost outstanding comprehensive performance, though an increase in time complexity occurs compared to DenseNet121 and DarkCovidNet. For the ratio per 10^6 pixels of the area in radar chart for each model, CrodenseNet, DenseNet121, DarkCovidNet, CvdNet and CovidNet are 1.108, 1.081, 1.007, 0.65 and 0.051, respectively. So comprehensive performance of CrodenseNet is similar with DenseNet121 and DarkCovidNet substantially, is much better than CvdNet and CovidNet.

6.2. Features of visualization

To assist clinicians in the analysis and diagnosis of COVID-19, the thermal map visualization technology Grade-CAM [31] is applied to highlight the feature areas that the model focuses on when classifying Chest X-ray images. As shown in Fig. 14, (a) and (c) are the original X-Ray images, while (b) and (d) are the corresponding thermal maps that show the key region/pixel of feature extraction. The more vibrant the color in the thermal map, the more attention the model pays. Except for the lung, the red regions on COVID-19 recognition are concentrated in the upper right and upper left of the image, including more shoulder regions or more noise pixels. It is because that the pathological features of COVID-19 are concentrated at the lung edge, and the extracted features of COVID-19 are concentrated at the lung edge, and the extracted features.
features cannot be completely concentrated throughout the lung. The red regions on Normal recognition are concentrated in the right lung, where there are fewer interference pixels. Because the lung of Normal class patient has no pathological features, the model can focus on the left and right lungs. The red regions on Viral recognition are concentrated in the bilateral lungs, indicating that the pathological features of Viral pneumonia are more visible, and the model can accurately identify the lesion areas. When predicting COVID-19 and viral pneumonia, CrodenseNet readily receives the influence of interference pixels other than lung pixels. As a result, X-Ray images should be clipped appropriately before being fed into classification model to remove interference pixels and improve attention on key features.

7. Conclusion

CrodenseNet model on the basis of DenseNet framework, is proposed in this paper. Two DenseNet blocks with different convolutional kernel sizes and parallel cross-dense connections are applied for extraction of the deep features of Chest X-ray images, and nonlinear transformation with one-sided soft thresholding is used to filter interference noise, further to implement the automatic 3-class identification, such as COVID-19, normal, and viral pneumonia. CrodenseNet has achieved a precision of 97.9 %, a recall of 96.7 %, and a F1-score of 97.3 % for COVID-19 detection according to cross-validation experiments on COVID-19x datasets. CrodenseNet outperforms various the-state-of-art models in classification accuracy and generalization performance, but has higher computation complexity than the light models. Therefore, we would like to optimize the architecture of model to decrease time complexity. Meanwhile, the spatial mechanism is required to improve

![Radar chart of each model performance.](image)

![Heatmap showing the parts of the input image (a), (c) that triggered the prediction (b), (d).](image)
the ability of feature extraction on covid-19 infection. In the future, we would like to design more efficient architecture for covid-19 detection, reduce the time complexity of model, and improve the sensitivity of prediction.

Funding
This research is funded by the National Natural Science Foundation of China (81973749).

Ethical approval
Not required.

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.

References
[1] M.L. Holshue, C. DeBolt, S. Lindquist, K.H. Lofy, J. Wiesman, H. Bruce, C. Spitters, K. Ericson, S. Wilkerson, A. Tural, G. Diaz, A. Cohn, LeAnne Fox, A. Patel, S. I. Gerber, L. Kim, S. Toeg, X. Lu, S. Lindstrom, M.A. Fallansch, W.C. Weldon, H. M. Biggs, T.M. Uyeki, S.K. Pillai, First Case of 2019 novel coronavirus in the United States, N Engl. J. Med. 382 (10) (2020) 929–936.
[2] D.S. Hui, E.I. Azhar, T.A. Madani, F. Nouni, R. Kock, O. Dar, G. Ippolito, T. D. Mchugh, Z.A. Memish, C. Drosten, A. Zumla, E. Petersen, The continuing 2019-nCoV epidemic threat of novel coronaviruses to global health — The latest 2019 novel coronavirus outbreak in Wuhan, China, Int. J. Infect. Dis. 91 (2020) 264–266.
[3] F. Sarbudbo, K. El Haji, A. Vidal-Balle, J. Bargay Lleonart, Neurological consequences of COVID-19 and brain related pathogenic mechanisms: A new challenge for neuroscience, Brain Behav. Immun. Health 19 (2022), 100399.
[4] S. Alaabali, A. Alghashm, S. Araj, M. Alsaed, M. Alrasheed, H. Tuyay Rauf, M. Arif, M. Abed mohammed, COVID-19 public sentiment insights: a text mining approach to the gulf countries, Comput. Mater. Continua 67 (2) (2021) 1613–1627.
[5] N. Kumar, M. Mohammed, K. Abdulkareem, R. Damasevicius, S. Mostafa, M. Maashi, S. Chopra, Artificial intelligence-based solution for sorting COVID-related medical waste streams and supporting data-driven decisions for smart circular economy practice, Process Saf. Environ. Protect. 152 (2021) 482–494.
[6] Z. Alyasseri, M. Al-betar, I. Doush, M. Awadallah, A. Abasi, S. Makhadmeh, O. Alomari, K. Abdulkareem, A. Adam, R. Damasevicius, M. Mohammed and R. Zitar, “Review on COVID-19 diagnosis models based on machine learning and deep learning approacher”, Expert Systems, vol. 39, no. 3, 2021.
[7] Hasoon, Jamal N., et al. “Covid-19 Anomaly Detection and Classification Method Based on Supervised Machine Learning of Chest X-Ray Images.” Results in Physics, vol. 31, 2021, pp. 105045, 2021.
[8] A.S. Al-Waisy, M. Abed Mohammed, S. Al-Fahdawi, M.S. Maashi, B. Garcia-Zapirain, K. Hamed abdulkareem, S.A. Mostafa, N. Manoj kumar, D. Le, COVID-DeepNet: hybrid multimodal deep learning system for improving COVID-19 detection in chest X-ray images, Comput. Mater. Cont. 67 (2) (2021) 2409–2429.
[9] A. Barshooi, A. Amirkhani, A novel data augmentation based on Gabor filter and convolutional deep learning for improving the classification of COVID-19 chest X-Ray images, Biomed. Signal Process. Control 72 (2022), 103326.
[10] S. Rajpal, N. Lakhyani, A.K. Singh, R. Kohli, N. Kumar, Using handpicked features in conjunction with resnet-50 for improved detection of covid-19 from chest X-ray images, Chaos, Solitons Fractals 145 (2021), 110749.
[11] P. Kedia, Anjum, and R. Katarya., CoNet-19: a deep learning model for the detection and analysis of COVID-19 patients, Appl. Soft Comput. 104 (2021), 107184.
[12] S. Karakasog, G. Leonidis, Lightweight deep learning models for detecting COVID-19 from chest X-ray images, Comput. Biol. Med. 130 (2021), 104181.
[13] T. Ozturk, M. Talo, E.A. Yildirim, U.B. Baloglu, O. Yildirim, U. Rajendra Acharya, Automated detection of COVID-19 cases using deep neural networks with X-ray images, Comput. Biol. Med. 121 (2020), 103792.
[14] 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.
[15] L. Wang, Z.Q. 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).
[16] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[17] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for Semantic Image segmentation, Computer Vision – ECCV (2018) 833–851.
[18] S. Santurkar, D. Tsipras, A. Ilyas, and A. M. ady “How does batch normalization help optimization?” 2018, arXiv:1805.11604.
[19] Y. Li and Y. Yian. “Convergence analysis of two-layer neural networks with relu activation,” 2017, arXiv:1705.09886.
[20] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
[21] M. Zhao, S. Zhong, X. Fu, B. Tang, M. Pecht, Deep residual shrinkage networks for fault diagnosis, IEEE Trans. Ind. Inf. 16 (7) (2020) 4681–4695.
[22] M. Jacquard, S. Babaie, M. Jadidi, CapsNet regularization and its conjugation with resnet for signature identification, Pattern Recognit. 120 (2021), 107851.
[23] S. Han, H. Choi, Understanding dropout as an optimization trick, Neurocomputing 298 (2020) 64–70.
[24] C. Han, Y. Lei, X. Xie, D. Zhou, M. Gong, Learning smooth representations with generalized softmax for unsupervised domain adaptation, Inf. Sci. 544 (2021) 415–426.
[25] J.P. Cohen P. Morrison L. Dao K. Roth T.Q. Duong M. Ghassemi COVID-19 image data collection Prospective Predictions Are the Future” 2020, arXiv:2003.11597.
[26] A. Chung, “A Figure 1 COVID-19 chest x-ray data initiative,” GitHub. [Online]. Available: https://github.com/agchung/Figure1-COVID-chestxray-dataset. [Accessed: 10-Nov-2021].
[27] A. Chung, “Actualmed COVID-19 chest x-ray dataset initiative,” GitHub. [Online]. Available: https://github.com/agchung/Actualmed-COVID-chestxray-dataset. [Accessed: 10-Nov-2021].
[28] T. Rahman, “Covid-19 radiography database,” Kaggle, 06-Mar-2021, [Online]. Available: https://www.kaggle.com/tawsifurrahman/covid19-radiography-database. [Accessed: 10-Nov-2021].
[29] Radiological Society of North America, “RSNA Pneumonia Detection Challenge,” Kaggle, [Online]. Available: https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data. [Accessed: 10-Nov-2021].
[30] K. He, J. Sun, Convolutional neural networks at constrained time cost. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
[31] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: visual explanations from deep networks via gradient-based localization. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[32] W. Maashi, S. Chopra, Artificial intelligence-based solution for sorting COVID-related medical waste streams and supporting data-driven decisions for smart circular economy practice, Process Saf. Environ. Protect. 152 (2021) 482–494.
[33] B. Garcia-Zapirain, K. Hamed abdulkareem, S.A. Mostafa, N. Manoj kumar, D. Le, COVID-DeepNet: hybrid multimodal deep learning system for improving COVID-19 detection in chest X-ray images, Comput. Mater. Cont. 67 (2) (2021) 2409–2429.