Transfer learning-based convolutional neural network for COVID-19 detection with X-ray images

Kevser Sahinbas[1], Ferhat Ozgur Catak[2]

[1]ISTANBUL MEDIPOL UNIVERSITY, MANAGEMENT INFORMATION SYSTEM, ISTANBUL, TURKEY; [2]SIMULA RESEARCH LABORATORY, FORNEBU, NORWAY

1. Introduction

The biggest pandemic we have experienced in this century has emerged with the COVID-19 outbreak. So far, more than three million people have been infected and more than 200,000 people have died in more than 200 countries [1]. Even if the disease is diagnosed with clinical findings by physicians, the detection of this disease, which spreads quickly, using data science methods at the early stage is critical for fighting the disease.

According to some studies, clinical findings of COVID-19 include fever, cough, sore throat, and radiographic images [2,3]. Clinical finding testing is accepted as the reference standard for diagnoses; however, it is time-consuming and results in a false-negative diagnosis. Moreover, COVID-19 test kits number are limited in hospitals. Therefore, quick alternative diagnostic techniques to combat COVID-19 spread among people, and treatment and care for patients are vital, critical issues to fight COVID-19. At this point, medical imaging such as X-ray and computed tomography (CT) images, which are used for chest radiological imaging, are familiar test techniques used for the early and quick diagnosis of COVID-19 and have an important role in treating this disease [4]. In radiology, much study has concentrated on the chest CT of COVID-19 [5,6]. According to strong advice, the first step in detecting COVID-19 is to arrive at a diagnosis with radiological images [7].

Specifically, convolutional neural networks (CNNs) are used in computer vision to analyze images in deep learning [43]. The classification of X-ray images needs a deep architecture to cope with the complicated structure of images. We propose that pretrained deep CNN models might be able to extract specific graphical features of

COVID-19 and provide a clinical diagnosis by using radiographical changes of COVID-19 in CT images. In this way, the model can save critical time in diagnosing the disease.

In this study, we used X-ray images of patients undergoing COVID-19 scanning, which can be accessed publicly on GitHub. We collected datasets from different hospitals; thus, we resized the pictures to be 256 × 22. Because of the insufficient number of examples, we enriched the dataset by adding image augmentation methods. In this way, we tended to improve the performance of the classification model. Evaluation and experiment of the proposed model were achieved in terms of 50 positive and 50 negative COVID-19 X-ray images for training and 20 positive and 20 negative images for testing phases. In this study, the five different architectures of well-known pretrained deep CNN models (VGG16, VGG19, ResNet, DenseNet, and InceptionV3) were used for transfer learning by employing X-ray images of COVID-19 patients for our proposed method. Each pretrained deep CNN models can analyze an X-ray image to distinguish a COVID-19 case and healthy group owing to their more in-depth structure. Our model achieves a classification accuracy of 80% for VGG16, which gives the best performance. We used an open public dataset of chest X-ray and CT images to enable COVID-19 detection. By showing our results in graphs and tables, we demonstrate the performance of the classification.

We present a brief summary of the related research in relation to transfer learning-based diagnostics of COVID-19 using chest X-rays as inputs. X-rays and CT images have been extensively applied to detect COVID-19. Public datasets containing COVID-19 X-ray images of infected patients are provided by the studies. According to some studies, combining chest radiological imaging with clinical findings may assist in the early detection of COVID-19 [8,9]. Ardakani et al. [10] proposed a COVID-19 diagnostic method by applying an artificial intelligence technique with 1020 CT images from 180 infected patients as the COVID-19 group and 86 other patients as the non-COVID-19 group. They used 10 familiar CNNs to separate COVID-19 infection from non-COVID-19 groups. The best performances was realized using the ResNet101 and Xception methods, which achieved 99.51% and 99.02% accuracy rates, respectively. They proved that ResNet101 is a highly sensitive model for diagnosing COVID-19 infections. Ucar et al. [11] presented a SqueezeNet model network design a COVID-19 diagnosis with Bayesian optimization additive in X-ray images. Ozturk et al. [12] presented a model for COVID-19 detection using chest X-ray images by implementing diagnostics for binary classification multiclass classification using a DarkNet model that applied 17 convolutional layers with 1125 images (125 COVID-19). Their model can help radiologists by the cloud system. Hemdan et al. [13] identified a new COVIDX-Net deep learning framework for helping radiologists to diagnose COVID-19 automatically in X-ray images. They apply seven well-known CNN models such as VGG19 and DenseNet using 25 confirmed positive COVID-19 patients to classify COVID-19 in X-ray images. Wang et al. [14] presented a deep CNN framework called COVID-Net to detect COVID-19 cases in chest X-ray images. They provide a CXR dataset including 13,800 chest radiography images across a 13,725-patient open access data platform to assist clinicians. Ioannis et al. [15] detected coronavirus

using CNN architectures that apply transfer learning with 224 images with confirmed COVID-19 in a dataset of X-ray images. They achieved an overall accuracy of 97.82%. Narin et al. [16] proposed CNN-based models such as pretrained ResNet50, InceptionV3, and InceptionResNetV2 to detect COVID-19 cases in chest X-ray images with 98% accuracy, in which the pretrained ResNet50 model was the best one. Song et al. [17] introduced a CNN-based CT diagnosis system to classify COVID-19 patients using chest CT scans of 88 patients diagnosed with COVID-19, with an area under the curve of 0.99. Wang et al. [14] presented proof-of-principle for the modified Inception transfer learning CNN model to classify COVID-19 cases by using 325 CT images with 85.2%. Xiaowei at al [18]. present early screening three-dimensional deep learning model to separate COVID-19 pneumonia from influenza-A viral pneumonia and irrelevant-to-infection groups using CT images with 86.7% accuracy. Xu et al. [19] provided a pretrained deep CNN ResNet model to detect COVID-19 patients using CT images with an 86.7% accuracy performance rate. Sethy and Behera [20] proposed a CNN model built on the pretrained ResNet50 with an support vector machine (SVM) classifier using X-ray images and demonstrated the best performance. Finally, other studies in this field have focused on X-ray and CT datasets related to COVID-19 detection [21,22].

The organization of the chapter is as follows: Section 1 contains the introduction, information about COVID-19 disease, the aims of this chapter and a brief summary of related studies with X-ray and CT using neural networks methods. Section 2 includes materials and methods, CNN, and transfer learning methods. Section 3 provides materials and methods, information about data preprocessing, the amount of data used for testing and training, and evaluation model performance results; and finally, Section 4 provides the conclusion.

2. Convolutional neural network

In this section, we will consider deep neural network analyzes and transfer learning methods. CNN is the most frequently used neural network class to analyze visual images in deep learning. CNN mainly contains many layers of neural networks, providing solutions especially for image and video recognition, classification, and analysis. A CNN architecture was designed with inspiration from the organization's visual cortex, similar to the connection model of neurons in the human brain [23]. Recently, learning from large scale datasets such as ImageNet has been effective in CNN's success. CNN basically consists of three main layers. These are the convolution layer, the pooling layer and the fully connected layer. Basically, convolutional and pooling layers provide the learning of the model, while the full connection layer provides the classification [24].

The convolutional layer is the main part of CNN architectures. In this layer, we can learn about the properties of the inputs. The feature map is created by applying high- and low-level filters in the input image. In general, in this layer, the sigmoid, ReLU, or

tanh function can be used as an activation function [24]. The mathematical equation of the convolution process is presented in Eq. (24.1). Whereas m and n in the equation indicate the dimensions of kernel $m \times n$, i and j indicate the matrix coordinate from which the convolution will be calculated:

$$S(i,j) = (I * K)(i,j) = \sum_{m}\sum_{n} I(m,n)K(i-m, j-n) \qquad 24.1$$

A pooling layer is generally applied between the convoluted layers. Its main purpose is to reduce the computational power required to process the model by reducing the size of the feature map. In addition, it provides effective training of the model by removing invariable and dominant features from the model. Although there are many pooling processes, maximum and average pooling layers are most commonly applied.

In the full connection layer, all neurons from the previous layer are connected to each neuron in this layer. Depending on the structure of the CNN architecture, the full link layer can be one or more. The output layer comes after the last full link layer. In classification studies, at this stage, using Softmax regression, output distributions are obtained by acquiring probability distributions for the output classes [24].

2.1   Transfer learning

Transfer learning is a strategy in machine learning in which information extracted by a CNN from given and related data is transported to solve a different but related problem [25]. Transfer learning is based on the principle of developing learning by using information obtained from a related task learned in a new task by transfer. Pretrained deep learning networks are already well-trained using other datasets, which could be refined to obtain high accuracy with a much smaller dataset; thus, it is preferred by researchers [26].

Pan and Yang [27] provided a comprehensive review of transfer learning. In transfer learning, the learning process is not started from scratch; rather, it is started with patterns learned while solving a different problem. In this way, previous learning is used and it avoids having to start the learning process from the beginning. Besides, it prevents time-consuming computations and enables the creation of a correct model [28].

It can be said that transfer learning is a design methodology for machine learning, not a machine learning model or technique. This design methodology commonly applies to pretrained models. These pretrained models are based on deep evolution neural networks. In deep learning, this method includes the initial training of a CNN for a classification problem using large-scale training datasets. Because a CNN model can learn to extract critical properties of the image, the availability of data for introductory training is an essential part of successful training. Depending on the CNN's capacity to recognize and select the most outstanding display features, it is evaluated whether this model is fit for transfer learning.

2.1.1   Visual Geometry Group Network (VGG)

The VGG-Net model was developed by Simonyan et al. [29] with a small convolution in the network. Although it is a simple model, its most significant difference compared with previous models is that it is widely applied to CNN models because of its more in-depth structure, followed by layers of associated double or triple convolution layers [29]. In previous models, the layers of sharing and convolution follow each other. Approximately 138 million parameters are calculated in this model [30].

VGG has a good representation of features for more than a million images (ImageNet dataset) from 1000 different categories. The model can function as a useful feature extractor for suitable new images. ImageNet dataset is able to extract related features from images, even new ones that do not exist or that might be in entirely different categories in the dataset. This provides the advantage of using pretrained models as an effective feature remover [29].

Fig. 24.1 indicates the architecture of the VGG16.

The VGG16 architecture uses three convolution filters with 13 convolution layers for feature extraction; each convolution layer follows a ReLU layer and has maximum pooling layers for sampling. It has three layers that are fully linked for classification, two of which serve as hidden layers, whereas the final classification layer consists of 1000 units representing image categories in the ImageNet database [29].

This structure simulates a larger filter while preserving the benefits of smaller filter sizes. VggNet has been shown to perform better using fewer parameters, especially compared with previous models. In addition, two ReLU layers were used instead of a single ReLU layer for two convolution layers. Because the spatial size of the input volumes in each layer decreases (the result of the convolution and partnering layers), the depth of the volumes increases owing to the increasing number of filters. It works well for both object classification and edge detection problems [30].
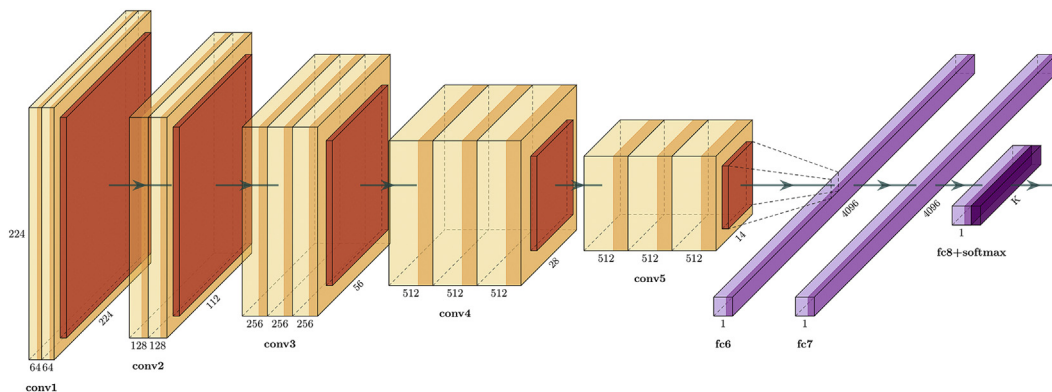

FIGURE 24.1 VGG16 architecture [29]. *cv*, convolution; *fc, fully connected*.

We use VGG16 network model to fine-tune our model. Suppose, we have a dataset with m samples $\{(x(1), y(1)), ..., (x(m), y(m))\}$ for training. The network overall cost function can be defined as [29]:

$$J(W, b) = \left[\frac{1}{m} \sum_{i=1}^{m} \left(\frac{1}{2} \left\| K_{w,b}(x^{(i)}) - y^{(i)} \right\|^2 \right) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_{l-1}} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} \left(W_{ji}^{(l)} \right)^2 \qquad 24.2$$

where $K_{w,b}\left(x^{(i)}\right)$ is the neural network model, $W_{ji}^{(l)}$ is the connection weight between the jth element of layer 1 and the ith element of layer $l+1$, and b is the bias term of the hidden layer neuron. Eq. (24.2) is a regulation item on the right side, which can prevent overfitting, reduces the weight greatly, and adjusts the relative importance of the two terms before and after cost function λ. Solving the minimum values of Eq. (24.2) and solving the minimum value of $J(W, b)$ adopts the recognized batch gradient descent optimization algorithm and calculates the partial derivative of W and b for the reverse conduction algorithm.

2.1.2 ResNet

ResNet is an architecture designed to be more in-depth structured than all previous architectures. It consists of 152 layers. ResNet was developed in 2015 [31]. It ranked first in the ImageNet competition held in 2015, with an error rate of 3.6% [31]. Fig. 24.2 demonstrates the architecture of the residual mapping of the model.

The most significant feature that distinguishes it from other architectures is that the blocks that feed the values to the next layers are added to the model. This value, added every two layers between the Linear and ReLU activation codes, changes the system value as stated. The value of $a^{[\int]}$ from the previous layer is added to value $a^{[\int +2]}$ [32].

Increasing the number of layers in a model normally means that the performance will increase, but in practice, the situation is changing. Thus, if $w[l+2] = 0$, according to the new theory, $a[l+2] = b[l+2]$. This causes a problem in which the derivative produces 0 value. It is undesirable [31]. However, now the value feed optimizes the learning error even if the value $a[l]$ from the two previous layers is 0, and the network is trained faster.
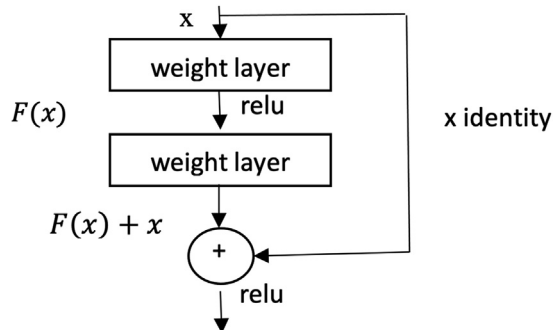

FIGURE 24.2 Residual mapping structure [31].

The architecture of ResNet50 consists of replacing both layer blocks in a 34-layer network with a three-layer bottleneck layer. This 50-layer network contains 3.8 billion floating point operations per second (FLOPs).

ResNet101 and ResNet152 build more layers than three-layer blocks use. ResNet152 contains 11.3 billion FLOPs. It currently has a lower complexity than VGG16/19 networks [31].

ResNet architecture was the winner in a study on the CIFAR10 dataset with 10 classes, 50,000 educational images, and 10,000 test images [33].

As the depth of CNN models increases, the problem of increasing gradients that vanish and explode affects the convergence of CNN models.

This architecture consists of residual blocks shown in Fig. 24.2. In the residual block, the convolution of input x yields an $F(x)$ result after the ReLU convolution series. This result is then added to the original x entry and is expressed as $H(x) = F(x) + x$.

The ResNet50 model provides easy training and a significant advantage of use owing to learning residuals from images instead of features [31].

2.1.3 DenseNet
While neural networks are being trained, feature maps decrease owing to convolution and subsampling processes. At the same time, there are losses in the image feature in transitions between layers. To use image information more effectively, Huang et al. [34] developed the DenseNet system.

In the system, each layer is fed forward to the other layers. In this way, any layer l can access the property information of all layers before it. It is possible to see the general network architecture of densely connected neural networks in Fig. 24.3 [34].
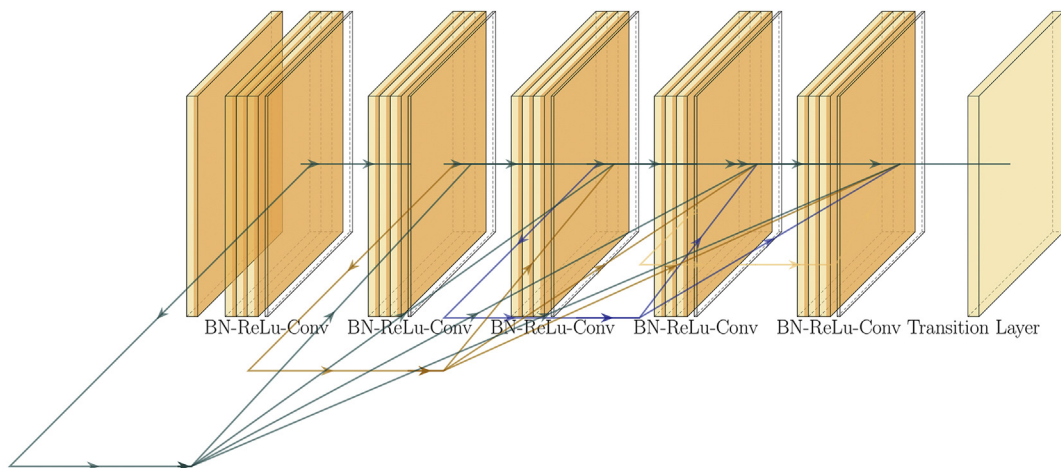
$$X_l = H_l(X_0, X_1, X_2, \ldots, X_{l-1})$$ 24.3

BN-ReLu-Conv   BN-ReLu-Conv   BN-ReLu-Conv   BN-ReLu-Conv   BN-ReLu-Conv   Transition Layer

FIGURE 24.3 DenseNet architecture [34]. *BN, batch norm; conv, convolution.*

Layers $[X_1, X_2, X_3, ..., X_{l-1}]$ are a combination of property information. H_l is the transfer function used to process property information. In this way, the rate of propagation of the features increases; it becomes easier to use again and the number of parameters decreases significantly.

2.1.4   InceptionV3

In 2013, Lin et al. [35] proposed an innovative solution to the calculation complexities of previous models with the 1×1 convolution process. The idea behind Inception is to provide an optimal local sparse structure in a convolutional vision network that can be approximated and covered by readily available dense components [36]. The basic idea of 1×1 convolution is to reduce the number of channels in the image. Thus, the number of parameters is also reduced. Normally 1×1 convolution does not affect the size of the matrix; however, if the input matrix is multichannel, the channel number of the output matrix of the 1×1 convolution process is equal to the number of channels of the 1×1 convolution filter applied.

The network model named Inception consists of several modules. In each module, different sizes of convolution and maximum partnering are applied. In the literature, GoogLeNet is also known as InceptionV1. Later versions called InceptionV2, InceptionV3 [37], and InceptionV4 [38] were also developed. In InceptionV3 architecture, there are two parts: feature extraction and classification.

The feature extraction section includes the CNN. InceptionV3 is a well-known architecture; the input of the network should be an image of 299×299 pixels. On the other hand, the classification section is fully connected and includes Softmax layers.

The start modules consist of a 3×3 maximum joint and 1×1, 3×3, and 5×5 convolution layers, as indicated in Fig. 24.4. After applying this Inception layer on the data from the previous layer, the network collects filters at the output and transfers them to the next layer. with the contribution of this module, both general and specific properties of an object may be discovered. In Fig. 24.4, the boxes with number 1 in parentheses is the Inlet and the top one is the output of the model.

The cost of calculation increases because there will be too many parameters during the convolution process in the Inception structure in Fig. 24.4. To overcome this problem, as shown in Fig. 24.5, 1×1 convolution was applied to dimension reduction.
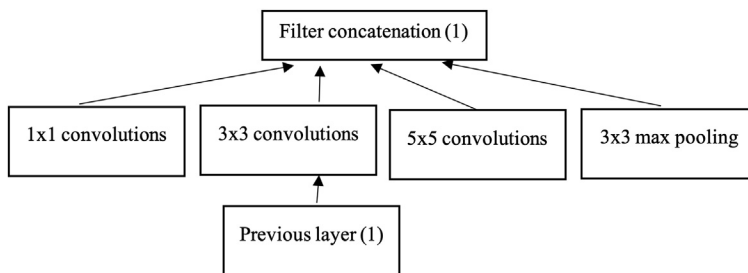

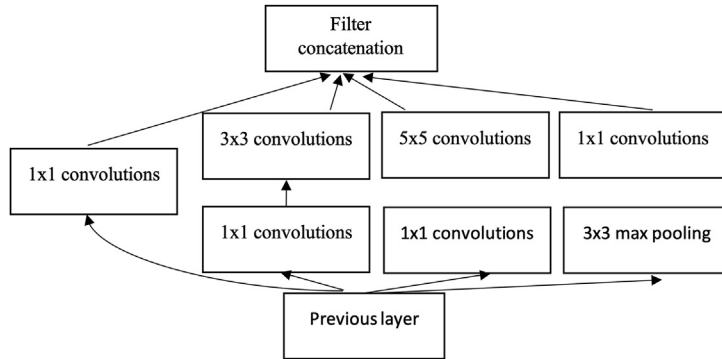**FIGURE 24.4** Inception module, naive version [36].

FIGURE 24.5 Inception module with dimension reductions [36].

3. Materials and method

3.1 Dataset

In this work, we used an open public dataset of chest X-ray and CT images of COVID-19 patients, which were positive or suspected for COVID-19 or other viral and bacterial types of disease (Middle East respiratory syndrome, severe acute respiratory syndrome, and acute respiratory distress syndrome [ARDS]). The dataset is collected from public sources as well as through indirect collection from hospitals and physicians. All images and data are published openly in the GitHub repository. The original dataset contains both X-ray images and CT scans. In this application, we chose the X-ray scan images to build a COVID-19 detection model. We used 50 positive and 50 negative COVID-19 posteroanterior X-ray scan images to create a training dataset and another 20 positive and 20 negative scans to test the model.

The X-ray scan images were rescaled to a size of 256×222. We implemented different image augmentation methods to produce more enhanced input instances, such as flipping right–left and up–down, rotation, and translation using five random angles.

Fig. 24.6 shows two examples from the dataset used in our experiments. The dataset used in the study can be said to contain insufficient X-ray scanning results. This dataset is the publicly accessible; it currently contains the most data and is tagged at the time of writing. In the future, both the number of datasets and the number of samples they contain will increase. Another issue is labeling. This dataset contains X-ray scan images containing the disease. There are only three records in the dataset with COVID-19–negative samples, and the "finding" field was chosen as no finding. Therefore, X-ray pictures with ARDS and *Streptococcus* results are tagged to select COVID-19–negative samples. In the future, we believe that samples of COVID-19–negative patients will increase in this type of data collection (Fig. 24.6).

We used image augmentation techniques to increase samples in the dataset and improve the performance of the classification model. Our image augmentation parameters were a rotation range of 20, zoom range of 15, width shift range of 0.2, height shift
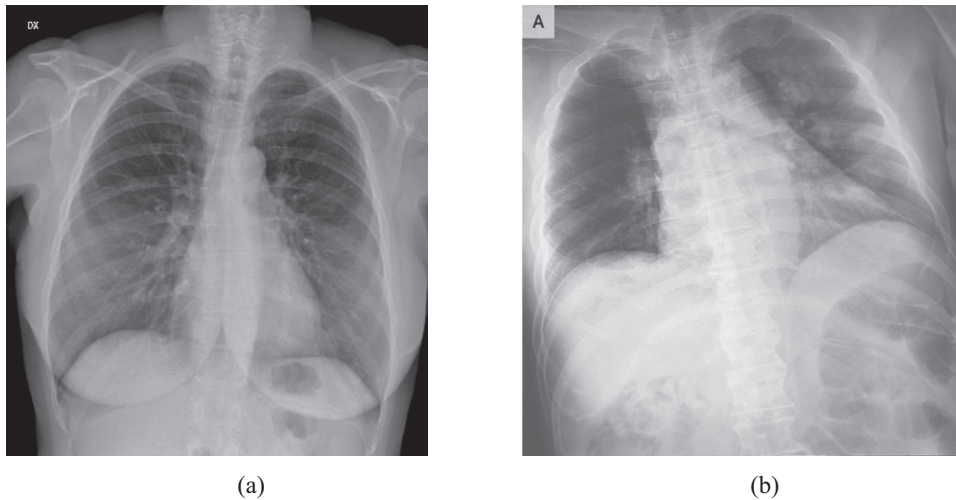
FIGURE 24.6 Example of an X-ray scan image taken from the dataset (A) with a label of COVID-19 negative; (B) with a label of COVID-19 positive.

range of 0.2, shear range of 0.15, and horizontal flipping. An example image augmentation technique is shown in Fig. 24.7.

3.2   Evaluation

Although the dataset used in our experiments is almost balanced, traditional accuracy-based performance evaluation is not enough to find an optimal classifier. We used four different metrics (overall prediction accuracy, average recall, average precision, and F_1 score) to evaluate classification accuracy. These are common measurement metrics in machine learning [39–41].

Precision is defined as the fraction of retrieved samples that are relevant:

$$\text{Precision} = \frac{\text{Correct}}{\text{Correct} + \text{False}} \qquad 24.4$$

Recall is defined as the fraction of relevant samples that are retrieved:

$$\text{Recall} = \frac{\text{Correct}}{\text{Correct} + \text{Missed}} \qquad 24.5$$

The F_1 score is defined as the harmonic mean of precision and recall:

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \qquad 24.6$$

The dataset was divided into two groups: 80% to train the model and 20% to evaluate the classification performance.
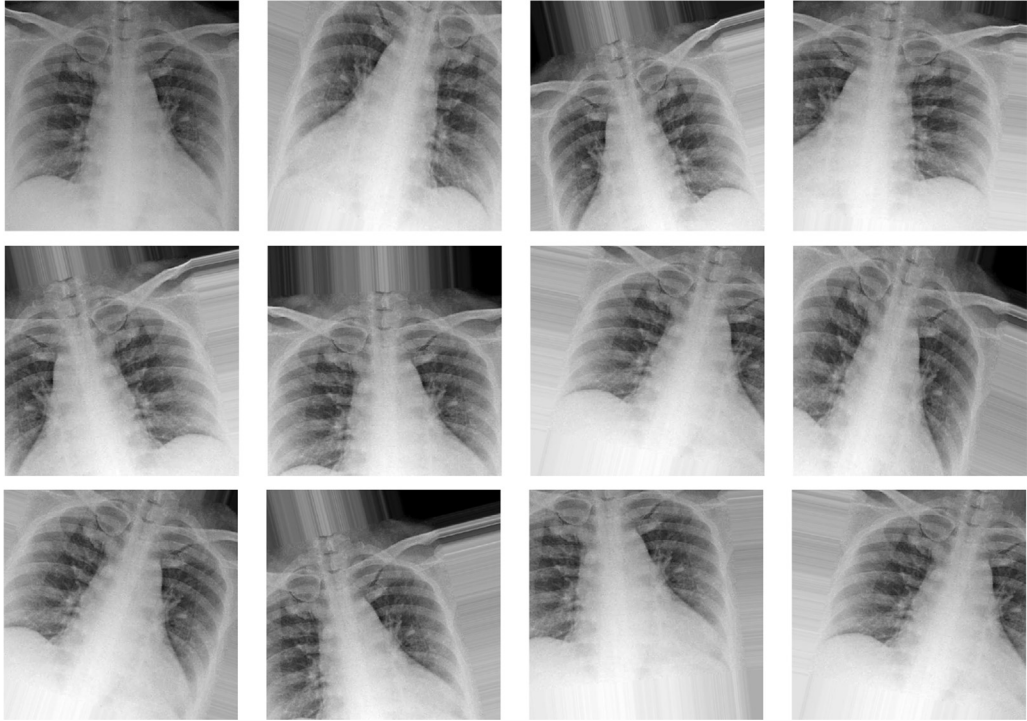
FIGURE 24.7 Data augmentation example. (A) Original image; (B–L) Augmented images.

3.3 Proposed model

We used the dataset containing X-ray images of patients undergoing COVID-19 scanning, which can be accessed publicly on GitHub [42]. Because these data were obtained from different hospitals, the picture resolutions differed from each other. To overcome this problem, we resized the pictures to 256×22. Because we used a CNN-based method, it was not affected by the adverse effects of this type of data compression. Owing to the insufficient number of examples, we enriched the dataset by adding image augmentation methods such as flipping and different angles. In this way, we aimed to improve the performance of the classification model.

In the last part of the transfer models we used, we added a heading model that we designed. In this way, we transformed the transfer learning models to detect COVID-19 patients from X-ray scanning images. The output of the classification model in this problem is in the form of binary classification such that it is positive and negative. We canceled optimization by training the layers belonging to the transfer models. The model we created is trained only by making parameter optimized to the heading model part. Fig. 24.8 shows the overall architecture.
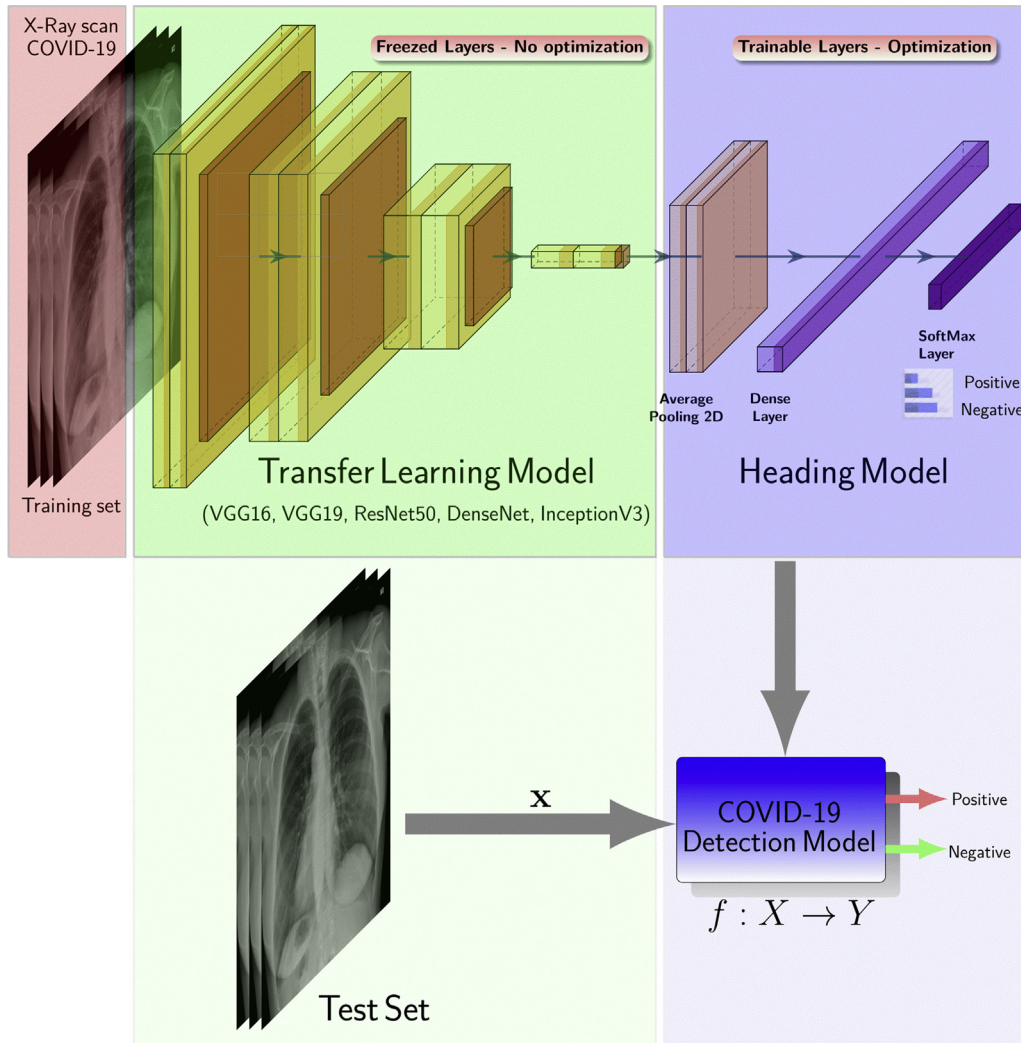
FIGURE 24.8 The proposed system models.

Fig. 24.9 shows the confusion matrix of each transfer learning model. VGG16 model has the best classification performance.

Findings in Table 24.1 indicate that the pretrained VGG16 model provides the highest classification performance of automated COVID-19 classification with 80% accuracy compared with the other four proposed models. The VGG19, DenseNet and InceptionV3 models indicate the same performance of classification with an accuracy of 0.60, and the ResNet shows the lowest classification performance with 0.50 accuracy.
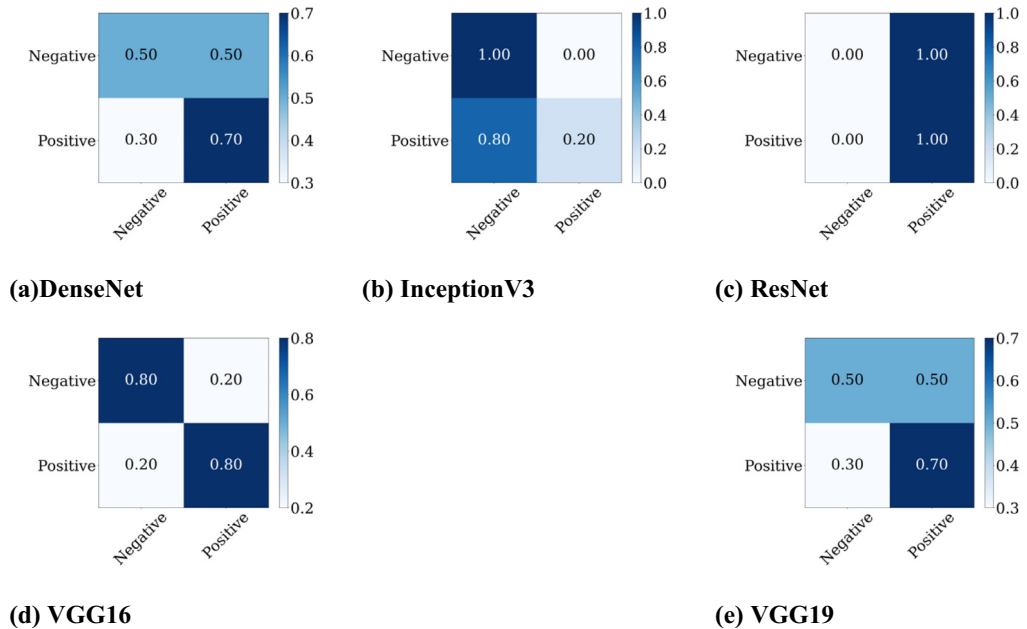
(a)DenseNet   **(b) InceptionV3**   **(c) ResNet**

(d) VGG16   **(e) VGG19**

FIGURE 24.9 The confusion matrix of each transfer model: (A) DenseNet; (B) InceptionV3; (C) ResNet; (D) VGG16; and (E) VGG19.

Table 24.1   Performance results for each convolutional neural network pr-trained model for COVID-19 of the proposed model.

Model	Accuracy	Precision	Recall	F_1 score
VGG16	0.80	0.80	0.80	0.80
VGG19	0.60	0.50	0.625	0.55
ResNet	0.50	0.0	0.0	0.0
DenseNet	0.60	0.50	0.625	0.55
InceptionV3	0.60	1.0	0.55	0.71

The VGG19, DenseNet, and InceptionV3 models have the same performance for COVID-19 classification with F_1 scores of 0.55. The VGG16 model presents the highest classification performance with an 80% F_1 score.

4. Conclusion

Medical imaging such as X-ray and CT images as rapid diagnosis techniques have a key role in treating COVID-19.

We first introduced the reason for using X-rays images in COVID-19 detection. We then described a few related studies on pretrained CNN methods using X-ray images. We used an open public dataset of chest X-ray and CT images to build COVID-19 detection. Because of the insufficient number of public COVID-19 datasets, we collected 50 positive and 50 negative COVID-19 X-ray images for training and 20 positive and 20 negative images for testing phases. We resized the pictures to be 256×22. We enriched the dataset by adding image augmentation methods such as flipping and different angles. We proposed five pretrained deep CNN models such as VGG16, VGG19, ResNet, DenseNet, and InceptionV3, which are employed for transfer learning by using the X-ray images of COVID-19 patients. The pretrained VGG16 model provided the highest classification performance of automated COVID-19 classification with 80% accuracy compared with the other four proposed models. By showing our results in graphs and tables, we indicated the classification performance. The VGG19, DenseNet, and InceptionV3 models had the same performance of classification with an accuracy of 0.60, and the ResNet had the lowest classification performance with 0.50 accuracy.

The dataset used in the study contains an insufficient number of examples. Therefore, the limited dataset of COVID-19 X-ray images was used. By increasing the number of instances in the dataset, the model can achieve more accurate performance.

References

[1] WHO - Coronavirus disease 2019 info web site n.d. https://www.who.int/emergencies/diseases/novel-coronavirus-2019 (accessed April 29, 2020).

[2] C. Huang, Y. Wang, X. Li, L. Ren, J. Zhao, Y. Hu, et al., Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China, Lancet 395 (10223) (2020) 497−506.

[3] T. Singhal, A review of coronavirus disease-2019 (COVID-19), Indian J. Pediatr. 87 (2020) 281−286.

[4] Z.Y. Zu, M.D. Jiang, P.P. Xu, W. Chen, Q.Q. Ni, G.M. Lu, L.J. Zhang, Coronavirus disease 2019 (COVID-19): a perspective from China, Radiology (2020), https://doi.org/10.1148/radiol.2020200490 (in press).

[5] S. Zhou, Y. Wang, T. Zhu, L. Xia, CT features of coronavirus disease 2019 (COVID-19) pneumonia in 62 patients in Wuhan, China, Am. J. Roentgenol. (2020) 1−8 (March).

[6] M. Chung, A. Bernheim, X. Mei, et al., CT imaging features of 2019 novel coronavirus (2019-nCoV), Radiology (February 2020) 200230, https://doi.org/10.1148/radiol.2020200230.

[7] Y. Li, L. Xia, Coronavirus disease 2019 (COVID-19): role of chest CT in diagnosis and management, Am. J. Roentgenol. (2020) 1−7, https://doi.org/10.2214/ajr.20.22954.

[8] W. Kong, P.P. Agarwal, Chest imaging appearance of COVID-19 infection, Radiol. Cardiothor. Imaging 2 (1) (2020) e200028.

[9] E.Y. Lee, M.Y. Ng, P.L. Khong, COVID-19 pneumonia: what has CT taught us? Lancet Infect. Dis. 20 (4) (2020) 384−385.

[10] A.A. Ardakani, A.R. Kanafi, U.R. Acharya, N. Khadem, A. Mohammadi, Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: results of 10 convolutional neural networks, Comput. Biol. Med. (April 30, 2020) 103795.

[11] F. Ucar, D. Korkmaz, COVIDiagnosis-Net: deep Bayes-SqueezeNet based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images, Med. Hypotheses 140 (July 2020) 109761.

[12] T. Ozturk, M. Talo, E.A. Yildirim, U.B. Baloglu, O. Yildirim, U.R. Acharya, Automated detection of COVID-19 cases using deep neural networks with X-ray images, Comput. Biol. Med. 121 (June 2020) 103792.

[13] E.E. Hemdan, M.A. Shouman, M.E. Karar, E.E.D. Hemdan, COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images, 2020 arXiv preprint arXiv:2003.11055.

[14] L. Wang, A. Wong, COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest Radiography Images, 2020 arXiv preprint arXiv:2003.09871.

[15] Ioannis D. Apostolopoulos1, Tzani Bessiana, COVID-19: Automatic Detection from X-Ray Images Utilizing Transfer Learning with Convolutional Neural Networks, 2003, p. 11617, arXiv.

[16] A. Narin, C. Kaya, Z. Pamuk, Automatic Detection of Coronavirus Disease (COVID19) Using X-Ray Images and Deep Convolutional Neural Networks, 2020 arXiv preprint arXiv:2003.10849.

[17] Y. Song, S. Zheng, L. Li, X. Zhang, X. Zhang, Z. Huang, Y. Chong, Deep Learning Enables Accurate Diagnosis of Novel Coronavirus (COVID-19) with CT Images, 2020 medRxiv.

[18] X. Xiaowei, J. Xiangao, M. Chunlian, D. Peng, L. Xukun, L. Shuangzhi, et al., Deep Learning System to Screen Coronavirus Disease 2019 Pneumonia, 2020, pp. 1−29. ArXiv 200209334.

[19] X. Xu, X. Jiang, C. Ma, P. Du, X. Li, S. Lv, et al., Deep Learning System to Screen Coronavirus Disease 2019 Pneumonia, 2020 arXiv preprint arXiv:200209334.

[20] P.K. Sethy, S.K. Behera, Detection of Coronavirus Disease (COVID-19) Based on Deep Features, 2020. https://www.preprints.org/manuscript/202003.0300/v1.

[21] M. Barstugan, U. Ozkaya, S. Ozturk, Coronavirus (COVID-19) Classification Using CT Images by Machine Learning Methods, 2020 arXiv preprint arXiv:2003.09424.

[22] C. Zheng, X. Deng, Q. Fu, Q. Zhou, J. Feng, H. Ma, X. Wang, Deep Learning-Based Detection for COVID-19 from Chest CT Using Weak Label, 2020. https://doi.org/10.1101/2020.03.12.20027185 medRxiv.

[23] S. Saha, Comprehensive Guide to Convolutional Neural Networks, 2018. Retrieved October 28, 2019, from, https://towardsdatascience.com/acomprehensive-guide-to-convolutional-neural-networks-the-eli5-way3bd2b1164a53.

[24] T. Guo, J. Dong, H. Li, Y. Gao, Simple convolutional neural network on image classification, in: 2017 IEEE 2nd International Conference on Big Data Analysis, 2017, pp. 721−724.

[25] M. Hussain, J.J. Bird, D.R. Faria, A study on CNN transfer learning for image classification, in: UK Workshop on Computational Intelligence, Springer, Cham, 2018, pp. 191−202. https://doi.org/10.1007/978-3-319-97982-3_16.

[26] Y. Chen, G. Tao, H. Ren, X. Lin, L. ve Zhang, Accurate seat belt detection in road surveillance images based on CNN and SVM, Neurocomputing 274 (2018) 80−87.

[27] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345−1359.

[28] W. Rawat, Z. Wang, Deep convolutional neural networks for image classification: a comprehensive review, Neural Comput. 29 (9) (2017) 2352−2449.

[29] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014, pp. 1409−1556. ArXiv Prepr. ArXiv.

[30] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European Conference on Computer Vision, 2013, pp. 818−833.

[31] K.M. He, X.Y. Zhang, S.Q. Ren, J. ve Sun, Deep Residual Learning for Image Recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CPVR), 2016, pp. 770−778.

[32] M.A. Kızrak, Derin Öğrenme ile Kalabalık Analizi Üzerine Detaylı Bir Araştırma, Bilişim Teknolojileri Dergzrakisi 11−3 (2018) 263−286.

[33] A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images, Tech Report, 2009.

[34] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700−4708.

[35] M. Lin, Q. Chen, S. Yan, Network in Network, 2013 arXiv preprint arXiv:1312.4400.

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1−9.

[37] C. Szegedy, V. Vanhoucke, et al., Rethinking the Inception Architecture for Computer Vision, 2015 arXiv: 1512.00567.

[38] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, February. Inception-v4, inception-resnet and the impact of residual connections on learning, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[39] A. Turpin, F. Scholer, User performance versus precision measures for simple search tasks, in: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Ser. SIGIR '06, ACM, New York, NY, USA, 2006, pp. 11−18 [Online]. Available, http://doi.acm.org/10.1145/1148170.1148176.

[40] C.D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, New York, NY, USA, 2008.

[41] J. Makhoul, F. Kubala, R. Schwartz, R. Weischedel, Performance measures for information extraction, in: Proceedings of DARPA Broadcast News Workshop, 1999, pp. 249−252.

[42] J.P. Cohen, P. Morrison, L. Dao, COVID-19 Image Data Collection. ArXiv (2020), arXiv:2003.11597.

[43] Ferhat Ozgur Catak, et al., Data augmentation based malware detection using convolutional neural networks, Peerj Computer Science 7 (2021) e346. https://doi.org/10.7717/peerj-cs.346. In this issue.