Contents lists available at ScienceDirect

Information Processing and Management

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

Detection of COVID-19 using deep learning techniques and classification methods

Çinare Oğuz, Mete Yağanoğlu [*]

Department of Computer Engineering, Faculty of Engineering, Ataturk University, Erzurum, Turkey

ARTICLE INFO

Keywords:
Deep learning
COVID-19
Classification
ResNet

ABSTRACT

Since the patient is not quarantined during the conclusion of the Polymerase Chain Reaction (PCR) test used in the diagnosis of COVID-19, the disease continues to spread. In this study, it was aimed to reduce the duration and amount of transmission of the disease by shortening the diagnosis time of COVID-19 patients with the use of Computed Tomography (CT). In addition, it is aimed to provide a decision support system to radiologists in the diagnosis of COVID-19. In this study, deep features were extracted with deep learning models such as ResNet-50, ResNet-101, AlexNet, Vgg-16, Vgg-19, GoogLeNet, SqueezeNet, Xception on 1345 CT images obtained from the radiography database of Siirt Education and Research Hospital. These deep features are given to classification methods such as Support Vector Machine (SVM), k Nearest Neighbor (kNN), Random Forest (RF), Decision Trees (DT), Naive Bayes (NB), and their performance is evaluated with test images. Accuracy value, F1-score and ROC curve were considered as success criteria. According to the data obtained as a result of the application, the best performance was obtained with ResNet-50 and SVM method. The accuracy was 96.296%, the F1-score was 95.868%, and the AUC value was 0.9821. The deep learning model and classification method examined in this study and found to be high performance can be used as an auxiliary decision support system by preventing unnecessary tests for COVID-19 disease.

1. Introduction

The COVID-19 pandemic has halted the economy around the world and has caused congestion in healthcare institutions, especially intensive care units (Firouzi et al., 2021; Fleuren et al., 2021; Garcia et al., 2021). It requires staying in intensive care in the later stages of the COVID-19 disease (Fleuren et al., 2021). It is tried to reach the diagnosis of COVID-19 by performing Polymerase Chain Reaction (PCR) test and rapid antigen test, by taking computed tomography (CT) image or X-ray image. CT images are created by combining X-ray images taken from multiple angles of the determined area of the patient's body. The PCR test result used to diagnose COVID-19 may be negative, especially in the early stage of the disease or when the viral load is low. Taking CT images from the patient at this stage plays an important role in the diagnosis of COVID-19 (Kang, Li, & Zhou, 2020). Fang et al. (2020) indicated may lead to negative PCR test result due to the immature development of nucleic acid detection technology, differences in detection rate from different manufacturers and inappropriate clinical sampling. Fang et al. (2020) found the sensitivity of the PCR test to be 71% in the early detection of COVID-19, and 98% of the sensitivity of the detection of COVID-19 by taking CT images. CT imaging is more preferred than X-ray imaging in the diagnosis of COVID-19. Because in the early stages of COVID-19 disease, the ground glass image in the lung may not be seen in X-ray images. The sensitivity of X-ray imaging is lower than that of CT imaging (Islam, Karray, Alhajj, & Zeng, 2021).

[*] Corresponding author.
E-mail addresses: cinare.oguz.91@gmail.com (Ç. Oğuz), yaganoglu@atauni.edu.tr (M. Yağanoğlu).

https://doi.org/10.1016/j.ipm.2022.103025
Received 6 December 2021; Received in revised form 1 July 2022; Accepted 2 July 2022
Available online 8 July 2022

Table 1
Summary table: COVID-19 detection studies using X-ray images.

Study	Method	Number of images	Data source	Performance
Ucar and Korkmaz (2020)	Bayes-SqueezeNet	1203 Normal 1591 Pneumonia 45 COVID-19	Open source	98.26 Accuracy
Panwar, Gupta, Siddiqui, Morales-Menendez, and Singh (2020)	nCOVNet	142 Normal 142 COVID-19	Open source	88% Accuracy
Oh, Park, and Ye (2020)	Patch-based CNN	8851 Normal 6012 Pneumonia 180 COVID-19	Open source	88.9% Accuracy
Mahmud, Rahman, and Fattah (2020)	CovXNet	1583 Normal 2780 Bacterial pneumonia 1493 Viral pneumonia 305 COVID-19	Open source	90.2% Accuracy
Panahi, Rafiei, and Rezaee (2021)	FCOD	505 Normal 435 COVID-19	Open source	96% Accuracy

This pandemic poses a challenge for healthcare workers around the world. It takes time to evaluate the test results of many patients. The need for solutions that can provide clinical support for the treatment of COVID-19 patients has increased (Romeo & Frontoni, 2022). Image analysis on medical images with decision support systems can provide accurate and rapid diagnosis of disease to cope with a large number of patient demands. For example, it may take up to 10 min for radiologists to manually evaluate CT images of a patient, while image analysis with decision support systems requires only a few seconds. COVID-19 was spreading very fast (Fleuren, Tonutti et al., 2021); Although it causes serious organ dysfunction such as pneumonia, kidney failure, sometimes it results in death. Therefore, early diagnosis of COVID-19 cases is necessary. Since the patient is not quarantined during the conclusion of the PCR test used in the diagnosis of COVID-19, the disease continues to spread.

2. Research objective

In this research, we created hybrid models by integrating deep learning models, which is a popular branch of machine learning, and traditional machine learning methods. Among the models created, a new high-performance COVID-19 detection system was developed with the hybrid model of ResNet-50 and SVM. In our proposed hybrid model framework, there are 2 modules including feature extraction with deep learning method and classification with traditional machine learning method.

We evaluated the proposed hybrid model on a new dataset of lung CT images obtained from a hospital setting and labeled by a radiologist. As shown in the experimental results, the proposed hybrid model (ResNet-50+SVM) achieved higher accuracy than the classical ResNet-50 method.

The main purpose of this article is to reduce the duration and amount of transmission of the disease by shortening the diagnosis time by taking CT images from patients who were admitted to the hospital due to the suspicion of COVID-19. In addition, it is aimed to provide a decision support system to radiologists in the detection of COVID-19.

3. Related works

In this section literature summary is discussed based on deep learning-based support systems for COVID-19 detection. The works in the literature were analyzed one by one based on the research methodologies and experimental results. These works are grouped according to the types of medical images used, such as CT and X-ray.

3.1. COVID-19 detection using X-ray images

A summary of studies using X-ray images in the literature is given in Table 1. Ucar and Korkmaz (2020) proposed a Bayes-SqueezeNet based COVIDiagnosis-Net for the diagnosis of COVID-19 on X-ray images. Dataset consisted of 45 COVID-19 cases, 1203 normal and 1591 non-COVID-19 pneumonia cases images. Dataset consists of a combination of three open-source datasets. According to the results obtained, the proposed model has an accuracy rate of 98.26% and an F1-score value of 98.25%.

Panwar et al. (2020) presented a deep learning neural network-based nCOVnet to detect COVID-19 patients on X-ray images. In the dataset, 142 X-ray images were selected as COVID-19 positive and 142 X-ray images obtained from the Kaggle dataset were selected as COVID-19 negative. 30% of this data set was used as test data. The proposed model detected COVID-19 positive images with an accuracy rate of 97%, while the overall accuracy was recorded as 88%.

Oh et al. (2020) proposed a patch-based CNN with few trainable parameters. The dataset includes X-ray images of 180 COVID-19, 6012 pneumonia and 8851 normal cases. The proposed model has an accuracy rate of 88.9% in the experiment.

Mahmud et al. (2020) used X-ray images as a data set for the detection of COVID-19 and other pneumonia cases. A deep learning model named CovXNet has been proposed. The data set consists of 305 COVID-19, 2780 bacterial pneumonia, 1493 viral pneumonia

Table 2
Summary table: COVID-19 detection studies using CT images.

Study	Method	Number of images	Data source	Performance
Wu et al. (2020)	COVNet	1325 Normal 1735 Pneumonia 1296 COVID-19	Multiple hospital environment	90% sensitivity
Zheng et al. (2020)	DeCOVNet	630	Multiple hospital environment	90.7% sensitivity
Singh et al. (2020)	MODE-CNN	75 Normal 75 COVID-19	Open source database	90.7% sensitivity
Pham (2020)	DenseNet-201	397 Normal 349 COVID-19	Open source database	96.2% accuracy
Wu, Chen, Zhong, Wang, and Shi (2021)	COVIDAL	342 Normal 316 Pneumonia 304 COVID-19	Open source database	86.6% accuracy
Rohila, Gupta, Kaul, and Sharma (2021)	ReCOV-11	1110	MosMedData	94.9% accuracy
Wang et al. (2021)	Modified Inception V3	1065	Multiple hospital environment	85.2% accuracy
Li, Yang, Liang, and Wu (2021)	CheXNet	397 Normal 349 COVID-19	Open source database	87% accuracy
Shui-Hua et al. (2022)	DRAPNet	306 Normal 281 Pneumonia 284 COVID-19 293 SPT	Multiple hospital environment	95.49% F1-Score
Wang, Zhang and Zhang (2021)	DSSAE	306 Normal 281 Pneumonia 284 COVID-19 293 SPT	Multiple hospital environment	92.32% F1-Score

and 1583 normal images. An accuracy rate of 97.4% was determined for the detection of COVID-19 images, and an accuracy rate of 90.2% for the general classification.

Panahi et al. (2021) proposed an inception architecture based Fast COVID-19 Detector for rapid detection of COVID-19 on X-ray images. The dataset consists of 940 open source X-ray images containing COVID-19 and normal cases. 80% of this data set is reserved for training and 20% for testing. The accuracy rate in detecting COVID-19 was 96%.

3.2. COVID-19 detection using CT images

A summary of studies using CT images in the literature is given in Table 2. Singh, Kumar, Kaur, et al. (2020) proposed a CNN for detect COVID-19 patients. The dataset consists of 75 COVID-19 positive and 75 COVID-19 negative CT images. Experiments were conducted using different variations while separating the data set. The best variation in the proposed model was found to be 90% for training and 10% for testing. The proposed model achieved 90.72% specificity, 90.7% sensitivity, 93.25% accuracy and 89.96% F1 score.

Wu et al. (2020) offered a model called COVNet based on the ResNet50 deep learning technique to reach the diagnosis of COVID-19. The data set consisting of CT images includes 4536 images, 1296 of which are COVID-19, 1735 are pneumonia and 1325 are normal. 90% of the dataset is reserved for training and 10% for testing. According to the experimental results, the proposed model achieved 90% sensitivity, 96% specificity and 96% AUC.

Zheng et al. (2020) proposed a 3D deep CNN type model (DeCoVNet) for the detection of COVID-19. The dataset consists of 630 CT images collected from the hospital environment. 80% of the data set is reserved for training and 20% for testing. In the experiments, 90.1% accuracy, 90.7% sensitivity, 91.1% specificity, 95.9% AUC value were obtained.

Wu et al. (2021) offered deep learning based COVID-AL. The dataset consists of CT images collected from the China Thorax CT Image Review Consortium (CC-CCII). In this data set, 304 COVID-19, 316 pneumonia and 342 normal case images were used. An accuracy rate of 86.6% was founded.

Pham (2020) proposed DenseNet-201 transfer learning model for COVID-19 detection. The performance of the model was evaluated on 746 CT images obtained from the open source dataset. The dataset contains 349 COVID-19 images and 397 normal images. Data augmentation was not applied. An accuracy value of 96.2% was obtained in the experimental results.

Wang, Kang et al. (2021) presented by changing the architecture of InceptionV3, which is the transfer-learning model for COVID-19 detection. The performance of the model was evaluated on 1065 CT images. In the experimental results, an accuracy value of 85.2% was obtained.

Li et al. (2021) proposed a pre-trained CheXNet model for COVID-19 detection. The performance of the model was evaluated on 746 CT images obtained from the open source dataset. The dataset contains 349 COVID-19 images and 397 normal images. In the experimental results, an accuracy value of 87% was obtained.

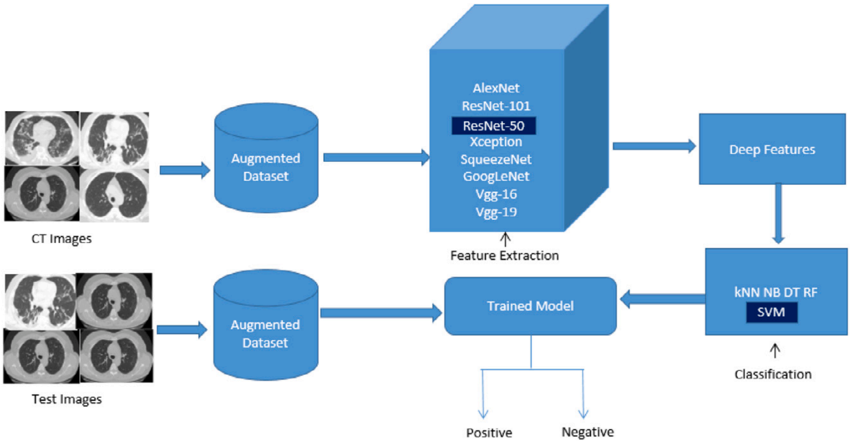

Fig. 1. Block diagram.

Shui-Hua et al. (2022) proposed a new deep rank-based mean pooling network (DRAPNet) model, based on the rank-based mean pooling module (NRAPM) and inspired by the VGG network, using CT images for COVID-19 detection. The performance of the model was evaluated on a multi-hospital dataset containing 1164 CT images. The dataset contains 284 COVID-19, 281 pneumonia, 293 pulmonary tuberculosis and 306 normal images. According to the test results, the proposed model reached an F1-score value of 95.49.

Rohila et al. (2021) proposed a model called ReCOV-101 for detection of COVID-19. The model is a DCNN model using ResNet-101. Preprocessing such as segmentation has been applied on full chest CT scans. In the experimental results, the accuracy value of 94.9% was reached.

Wang, Zhang et al. (2021) proposed a deep-stack sparse auto encoder (DSSAE) model for COVID-19 detection. Features are obtained using two-dimensional fractional Fourier entropy. The performance of the model was evaluated on a multi-hospital dataset containing 1164 CT images. The dataset includes 284 COVID-19, 281 pneumonia, 293 s pulmonary tuberculosis(SPT) and 306 normal images. According to the experimental results, the proposed model achieved an F1-score value of 92.32. In addition, using fusion of CT and X-ray images were obtained by Zhang, Zhang, Zhang, and Wang (2021). They proposed a multi-input deep convolutional attention network (MIDCAN) model for COVID-19 detection. The performance of the model was evaluated on a multi-hospital dataset containing mixed CT and X-ray images. The dataset contains 42 COVID-19 and 44 normal images. According to the experimental results, the proposed model achieved an accuracy value of 98.02%.

In the detection of COVID-19 disease, experts prefer CT imaging more than X-ray imaging. Because in the early stages of COVID-19 disease, the ground glass image in the lung may not be seen in X-ray images. The sensitivity of X-ray imaging is lower than that of CT imaging (Islam et al., 2021). Therefore, CT images were used in this study. In the studies in the literature, deep learning models have been used alone. In this study, performance was increased by using deep learning models and classification methods together. Also the run times of the methods used in most of the studies in the literature are not mentioned.

4. Material and method

The block diagram of this study is shown in Fig. 1. Patients who come to Siirt Research and Training Hospital with the suspicion of COVID-19 between May 2020 and June 2020 are accepted as participants. Deep features are extracted from different deep learning models (ResNet-50, ResNet-101, AlexNet, Vgg-16, Vgg-19, GoogLeNet, SqueezeNet and Xception) by using the images reserved for training from the dataset. By using these deep features in various classification methods (SVM, kNN, RF, DT and NB), it is tried to find the deep learning model and classification method pair with the highest performance. In this study, the fully connected layer of all deep learning models is replaced with a new fully connected layer whose output number is equal to the number of classes in the new dataset. In the classification process, it was done with various traditional machine learning methods instead of softmax.

In the input layer of the deep learning model, images are adapted to the model. In the convolution layer, feature maps are created with the convolution filters of the model. In the first convolution layer, the edges of the input are calculated. The edges show the high-frequency regions of the input. After these calculations, the padding method is applied because the output size will be different from the input size. Thus, the output and input sizes are synchronized. Deep features are extracted from the fc layer of the model. Learning takes place by using the deep features and class labels of the training data in different classification methods. Then, the deep features of the images left for testing from the dataset are extracted with deep learning models. The estimation process is performed by using the extracted deep features in the trained classification methods. Finally, by comparing the class labels of the test data with the predicted class labels, deep learning models and classification methods are evaluated with performance metrics such as accuracy, sensitivity, specificity, F1 score, runtime and AUC.

Table 3
Details of dataset.

	COVID-19 positive	COVID-19 negative
Training set	485	590
Test set	122	148
Total	607	738

4.1. Data set

Ataturk University Non-Invasive Clinical Research Ethics Committee approval was obtained for the data set used in this study. With this approval, COVID-19 positive and COVID-19 negative CT data from patients who applied to Siirt Education and Research Hospital COVID-19 Diseases polyclinics between May 2020 and June 2020 were obtained through the automation system and a data set was created. CT images were taken from adults over 18 years of age. Data set was obtained from 1275 patients in total. 53% of patients who are positive for COVID-19 consist of male patients. One hundred of these patients are intensive care patients, and CT images were taken again from the same patient on one day, seven days and 14 days, depending on the length of stay for each patient. 72% of intensive care patients consist of patients over 54 years of age. A data set was created using a total of 1345 CT images. While 738 of them were negative for COVID-19, 607 of them were positive for COVID-19. In the study, 270 (20%) of the CT images in the data set were used for testing. Details of the dataset are given in Table 3.

4.2. Deep learning models

4.2.1. AlexNet

Maximum pooling(max pool) is performed on 3 × 3 pixel windows. For maximum pooling, the stride is set to [2 2], padding is set to [0 0 0 0]. The activation function (RELU) is applied after each convolutional layer (conv layer) and to the output of the fully connected layer (fc layer). The first conv layer contains convolution filters with 96 cores of 11 × 11 size. In the first conv layer, the stride for the filter is set to [4 4] and padding is set to [0 0 0 0]. Normalization and pooling operations are applied to the outputs of the first and second conv layers. No pooling or normalization is applied between the third, fourth, and fifth conv layers. The second conv layer contains convolution filters with 256 cores of 5 × 5 size. In the second conv layer, the stride for the filter is set to [1 1] and padding is set to [2 2 2 2]. The third conv layer contains convolution filters with 384 cores of 3 × 3 size. In the third conv layer, stride for the filter is set to [1 1] and padding is set to [1 1 1 1]. The fourth conv layer has convolution filters with 384 cores of 3 × 3 size. In the fourth conv layer, the stride for the filter is set to [1 1] and padding is set to [1 1 1 1]. The fifth conv layer has convolution filters with 256 cores of 3 × 3 size. In the fifth conv layer, the stride for the filter is set to [1 1] and padding is set to [1 1 1 1]. The first two fc layers each contain 4096 channels and the last fc layer contains 1000 channels.

4.2.2. ResNet

The image input size of the model is 224 × 224. Maximum pooling is performed on 3 × 3 pixel windows. For maximum pooling, the stride is set to [2 2], padding is set to [1 1 1 1]. The 7 × 7 average pooling is done after the fifth conv layer. RELU is implemented after each conv layer. The first conv layer contains convolution filters with 64 cores of 7 × 7 size. In the first conv layer, the stride for filters is set to [2 2] and padding is set to [3 3 3 3]. The second conv layer takes the output of RELU and pooling operations after the first conv layer as input. It includes convolution filters with 64 cores of 1 × 1 and 3 × 3 sizes and 256 cores of 1 × 1 size. In the second conv layer, the stride for filters is set to [1 1], padding is set to [0 0 0 0] for 1 × 1 filters and 'same' for 3 × 3 filters. The third conv layer takes the output of RELU and normalization operations after the second conv layer as input. It includes convolution filters with 128 cores of 1 × 1 and 3 × 3 sizes and 512 cores of 1 × 1 size. In the third conv layer, the stride for filters is set to [1 1], padding is set to [0 0 0 0] for 1 × 1 filters and 'same' for 3 × 3 filters. The fourth conv layer takes the output of the RELU and normalization operations after the third conv layer as input. It includes convolution filters with 256 cores of 1 × 1 and 3 × 3 sizes and 1024 cores of 1 × 1 size. In the fourth conv layer, the stride for filters is set to [1 1], padding is set to [0 0 0 0] for 1 × 1 filters and 'same' for 3 × 3 filters. The fifth conv layer takes the output of the RELU and normalization operations after the fourth conv layer as input. It includes convolution filters with 512 cores of 1 × 1 and 3 × 3 sizes and 2048 cores of 1 × 1 size. In the fifth conv layer, the stride for filters is set to [1 1], padding is set to [0 0 0 0] for 1 × 1 filters and 'same' for 3 × 3 filters.

4.2.3. Vgg

Vgg takes 224 × 224 image input. There are five max pool filters placed between conv layers for size reduction. Maximum pooling is performed on 2 × 2 pixel windows. For max pool, the stride is set to [2 2], padding is set to [0 0 0 0]. In the dropout layer, the outputs are reduced by 50% (drop value 0.5).

The last layer is the softmax layer. The stride for filters in conv layers is set to [1 1] and padding is set to [1 1 1 1]. The first conv layer contains convolutional filters with 64 cores of 3 × 3 size. The second conv layer takes the output of RELU and pooling operations after the first conv layer as input. Includes convolution filters with 128 core of 3 × 3 size. The third conv layer takes the output of RELU and pooling operations after the second conv layer as input. It includes convolution filters with 256 cores of 3 × 3 size. The fourth conv layer takes the output of RELU and pooling operations after the third conv layer as input. It includes convolution filters with 512 core of 3 × 3 size. The fifth conv layer takes the output of RELU and pooling operations after the fourth conv layer as input. Includes convolution filters with 512 core of 3 × 3 size. After the fifth conv layer, the final pooling takes place. Its output is given to the fc layer. RELU and dropout operations are applied between fc layers. The second conv layer takes the output of RELU and pooling operations after the first conv layer as input. This layer contains convolution filters with 128 cores of 3 × 3 size. The third conv layer takes the output of RELU and pooling operations after the second conv layer as input. This layer contains convolution filters with 256 cores of 3 × 3 size. The fourth conv layer takes the output of RELU and pooling operations after the third conv layer as input. This layer contains convolutional filters with 512 cores of 3 × 3 size. The fifth conv layer takes the output of RELU and pooling operations after the fourth conv layer as input. This layer contains convolutional filters with 512 cores of 3 × 3 size. After the fifth conv layer, the final pooling takes place. Its output is given to the fc layer. RELU and dropout operations are applied between fc layers.

4.2.4. GoogLeNet

The most obvious difference from the GoogLeNet deep learning model from other models is the Inception module. The basic logic of the Inception module is to reasonably reduce the size where the computational requirements would increase too much. Size reduction is performed with filter elements of different sizes such as 1 × 1, 3 × 3, 5 × 5 in this module.

GoogLeNet's image input size is 224 x 224. An alternative parallel max pool path is added at each stage. A total of five pooling processes are applied. Maximum pooling is performed on 3 × 3 pixel windows. For max pool operations, the stride is set to [2 2], padding is set to [0 1 0 1]. The first conv layer contains convolution filters with 64 cores of 7 × 7 size. In the first conv layer, the stride is set to [2 2] for the filters and the padding is set to [3 3 3 3]. The second conv layer takes the output of RELU, maximum pooling and normalization operations after the first conv layer as input. It includes convolution filters with 64 cores of size 1 × 1 and 192 cores of size 3 × 3. In the second conv layer, the stride is set to [1 1] for 1 × 1 filters, padding is set to [0 0 0 0] and the stride is set to [1 1] for 3 × 3 filters, padding is set to [1 1 1 1]. After the second conv layer, RELU takes the output of max pool and normalization operations as input to the inception module. After the second conv layer comes 9 inception modules. In Inception modules, the stride is set to [1 1], padding is set to [0 0 0 0] for 1 × 1 filters, [1 1 1 1] for 3 × 3 filters, and [2 2 2 2] for 5 × 5 filters. Maximum pooling is applied to the output of the second inception module and the output of the seventh inception module. The fc layer takes the output of 7 × 7 average pooling and dropout (0.4 drop value) operations after the last inception module as input.

4.2.5. SqueezeNet

Its different side from other deep learning models is fire modules. This module consists of an expansion layer consisting of a mixture of 1 × 1 and 3 × 3 convolution filters and a compression conv layer with only 1 × 1 convolution filters (Iandola et al., 2016). The image input size of the network is 227 × 227 (Bhole & Kumar, 2020).

The SqueezeNet architecture starts with the independent conv layer. Then, the maximum pooling process takes place in the size of 3 × 3. For maximum pooling, the stride is set to [2 2], padding is set to [1 1 1 1]. Next comes 8 fire modules and conv layers. Two more max pool operations occur between fire modules. The stride is set to [2 2] for max pool between fire modules and padding is set to [0 1 0 1]. Finally, it ends with the average pooling process. In the dropout layer, the outputs are reduced by 50% (0.5 drop value).

The first conv layer contains convolution filters with 64 cores of 3 × 3 size. For filters in the first conv layer, the stride [2 2] is set to padding is set to [0 0 0 0]. The stride is set to [1 1] for filters in fire modules, padding is set to [0 0 0 0] for filters of 1 × 1 size and [1 1 1 1] for filters of 3 × 3 size. The second conv layer takes the output of the dropout process after 8 fire modules as input. It contains convolution filters with 1000 cores of 1 × 1 size. In the second conv layer, the stride is set to [1 1] for the filters, padding is set to [0 0 0 0]. The average pooling layer takes the output of the RELU operation after the second conv layer as input.

4.2.6. Xception

The image input size of the network is 229 x 229 (Chollet, 2017). The model includes 14 block conv layers. Maximum pooling is performed on 3 × 3 pixel windows. For maximum pooling operations, stride is set to [2 2], padding is set to 'same'. The first block conv layer contains convolution filters with 32 cores of 3 × 3 size and 64 cores of 3 × 3 size. In the first block conv layer, the stride is set to [2 2] for 32-core filters, the stride is set to [1 1] for 64-core filters, and padding is set to [0 0 0 0]. The second block conv layer takes the output of the RELU and normalization operations after the first conv layer as input. It includes convolution filters with 64-core of 3 × 3 size and 128-core of 1 × 1, 3 × 3 sizes. In the second block conv layer, stride is set to [1 1] for 3 × 3 filters, padding is set as 'same', stride is set as [1 1] for 1 × 1 filters, padding is set as [0 0 0 0]. The third block conv layer takes the output of RELU, normalization and pooling operations after the second block conv layer as input. It includes convolution filters with 128 cores of 3 × 3 size and 256 cores of 1 × 1, 3 × 3 sizes. In the third block conv layer, the stride is set to [1 1] for 3 × 3 filters, padding is set as 'same', stride is set to [1 1] for filters of 1 × 1 size, padding is set as [0 0 0 0]. The fourth block conv layer takes the output of RELU, normalization and pooling operations after the third block conv layer as input. It includes convolution filters with 256 cores of 3 × 3 size and 728 cores of 1 × 1, 3 × 3 sizes. In the fourth block conv layer, stride is set to [1 1] for 3 × 3 sized filters, padding is set as 'same', stride is set to [1 1] for 1 × 1 sized filters, padding is set to [0 0 0 0]. The fifth block conv layer takes the output of RELU, normalization and pooling operations after the fourth block conv layer as input. It includes convolution filters with 728 cores of 1 × 1 and 3 × 3 sizes. In the fifth block conv layer, stride is set to [1 1] for 3 × 3 size filters, padding is set as 'same', for 1 × 1 size filters stride is set as [1 1], padding is set as [0 0 0 0]. The sixth block conv layer takes the output of RELU and normalization operations after the fifth block conv layer as input. It includes convolution filters with 728 cores of 1 × 1 and 3 × 3 sizes. In the sixth block conv layer, the stride is set as [1 1] for 3 × 3 filters, padding is set as 'same', stride is set as [1

1] for filters of 1 × 1 size, padding is set as [0 0 0 0]. Until the thirteenth block conv layer, the other block conv layers perform the same operations and contain convolution filters of the same size. The thirteenth block conv layer takes as input the RELU and normalization operations after the twelfth block conv layer. It includes convolution filters with 728 cores of 3 × 3 size and 1024 cores of 1 × 1, 3 × 3 sizes. In the thirteenth block conv layer, the stride is set to [1 1] for 3 × 3 filters, padding is set as 'same', for 1 × 1 size filters, stride is set to [1 1], padding is set as [0 0 0 0]. The fourteenth block conv layer takes the output of the RELU, normalization and pooling operations after the thirteenth block conv layer as input. It includes 1536 cores of 1 × 1, 3 × 3 sizes and convolution filters with 2048 cores of 1 × 1 size. In the fourteenth block conv layer, stride is set to [1 1] for 3 × 3 filters, padding is set as 'same', stride is set as [1 1] for 1 × 1 size filters, padding is set as [0 0 0 0].

4.3. Classification methods

4.3.1. k nearest neighbor classification method

According to the k Nearest Neighbor (kNN) algorithm, the number of neighbor's k is determined. The number of neighbors is not fixed. The choice of the best k varies according to the data. In binary classifications, it is better to choose k as an odd number to avoid binding decisions. In the data set in our study, the k value was tested for all values from 1 to 10 and the best result was determined as "k = 5". The distance function is calculated between the samples in the training set and the test data. The Euclidean method is used to calculate the distance function (Hu, Huang, Ke, & Tsai, 2016). The distance function is given in Eq. (1). Then the obtained distances are sorted in ascending order and the first k samples are selected. The most common class among these examples is considered to be the class of test data.

$$d(j,i)=\sqrt{\sum_{k=1}^{n}(x_{ik}-x_{jk})^2} \tag{1}$$

4.3.2. Support vector machines

Support Vector Machines (SVM) are the separation and classification of points on the plane by a line or hyperplane. While creating the SVM decision surface, it tries to maximize the distance between the two classes. Between these planes, there is only one hyperplane with a maximum boundary. The points that constrain the boundary width are called support vectors. The support vector algorithm tries to minimize the training error by classifying with the separating hyperplane with the largest boundary width. It aims to find a linear discriminating function with the largest margin that separates the classes from each other. For non-linearly separable samples, the samples can be separated linearly into another higher-dimensional. A decision boundary is created with the training data set and the test data is classified according to this boundary (Zhou, 2021).

4.3.3. Decision trees

The representation of the information presented in the Decision Trees (DT) can be easily understood. DT not only shows decisions but also contains explanations of decisions. DT starts at the root of the tree to guess the class label of an instance. Root Node: It represents the entire population. They were further divided into two or more homogeneous clusters. Initially, the entire training set is considered the root. Root property values are compared with the properties of the sample. That is, to follow the branch corresponding to this value and jump to the next node. Samples are distributed recursively based on property values. While the features are placed in the tree, determining them as root or internal nodes is done using some statistical approaches. These approaches are; entropy is a measure of randomness (Charbuty & Abdulazeez, 2021). When the probability is 0 or 1, the result is certain. Since there is no randomness, the entropy value is zero. The entropy calculation for a single feature is calculated as in Eq. (2).

$$E(S)=\sum_{i=1}^{c}p_i log_2 p_i \tag{2}$$

S is the current sample, c is the number of classes, and pi is the probability that the sample belongs to class i. The entropy calculation for multiple features is calculated as in Eq. (3). T is the current feature and X is the selected feature.

$$E(T,X)=\sum_{c\epsilon X}P(c)E(c) \tag{3}$$

Information gain is a measure of how well a particular feature discriminates against target classifications (Yuvaraj et al., 2021). Information gain and entropy are inversely proportional. The information gain is calculated based on the given feature values, the difference between the entropy before the split and the average entropy after the data set to split. Information gain is calculated as in Eq. (4).

$$IG(T,X)=E(T)-E(T,X) \tag{4}$$

These approaches calculate values for each feature. Values are sorted and features are placed in the tree in order. In other words, the feature with the highest information gain and the lowest entropy is placed at the root.

4.3.4. Random forest

Random Forest (RF) methods, one classifier instead of multiple generating many classifiers and then from their predictions learning that classifies new data with the votes received are algorithms. The RF method is considered as collective learning. In other words, more accurate results are obtained because it finds the average by using the results of more than one decision tree. When performing Random Forests based on classification data, the Gini index is usually calculated. GINI index measures class homogeneity. It is used to decide how the nodes in the Decision Tree branch are knotted. The Gini index formula is given in Eq. (5). Here, i represents the probability that the sample belongs to the class, and c represents the number of classes (Yang, Liu, Liu, & Li, 2021).

$$Gini = 1 - \sum_{i=1}^{c} (pi)^2 \tag{5}$$

4.3.5. Naive Bayes

The Naive Bayes (NB) method considers all the features to be equally important and independent from each other. It is based on Bayes' Theorem. In the NB method, $X = \{x_1, x_2, ..x_m\}$. X is an example from a test data set with m features. Let C_1, C_2,…, C_n be possible class values (Jahanbani Fard, 2015). In Eq. (6), the probabilities of the features according to the classes are calculated.

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) \tag{6}$$

In Eq. (7), the probability of each class is calculated to find out which class the data sample belongs to. The class with the highest result is designated as the class of the data sample.

$$P(C_i \mid X) = \frac{P(X \mid C_i)P(C_i)}{P(X)} \tag{7}$$

5. Experimental results

Recently, deep learning and traditional machine learning have become very popular in many disease diagnostic applications using medical images. Deep learning models have been developed for the diagnosis of COVID-19 using both CT and X-ray images (Ardakani, Kanafi, Acharya, Khadem, & Mohammadi, 2020; Oh et al., 2020; Singh et al., 2020; Ucar & Korkmaz, 2020; Wu et al., 2020). Traditional machine learning methods are being used actively for COVID-19 detection, prognosis and epidemic prediction and has contributed to greatly reduce the severity of the epidemic (Islam et al., 2021). In this study, the most commonly used classification methods in the context of disease diagnosis via medical images and eight pre-trained deep learning models were used to distinguish COVID-19 positive cases from the COVID-19 negative group. The study was carried out on MATLAB R2021a on a computer with Intel(R) Core (TM) i7-8550U CPU @ 1,80 GHz.16 GB RAM. 20% of the data set was used for testing.

Performance Metrics were calculated by creating a confusion matrix. Accuracy, sensitivity, specificity, precision and F1-score are calculated using equations 8, 9, 10, 11, 12 (Bozkurt, 2021).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{9}$$

$$Specificity = \frac{TN}{TN + FP} \tag{10}$$

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$F1 - score = 2 * \frac{Sensitivity * Precision}{Sensitivity + Precision} \tag{12}$$

TP: The image is estimated to be COVID-19, and this image is indeed COVID-19. TN: The image is estimated to be normal and this image is normal. FP: The image is estimated to be COVID-19, but this image is normal. FN: The image is estimated to be normal, but this image is COVID-19 (Afify & Zanaty, 2021).

The ROC curve is plotted by comparing the false positive rate (1-Specificity) with the correct positive rate (Sensitivity) at certain threshold settings. The area covered by the ROC curve gives the AUC value. It tells how much the model can distinguish between classes.

Table 4
Experimental results of ResNet-101.

	Accuracy (%)	Sensitivity (%)	Specificity (%)	F1L-score (%)	Precision (%)
NB	82.963	83.607	82.432	81.6	79.688
SVM	90.37	86.066	93.919	88.983	92.105
kNN	84.815	83.607	85.811	83.266	82.927
DT	76.296	70.492	81.081	75.439	72.882
RF	91.185	90.164	93.243	90.909	91.667

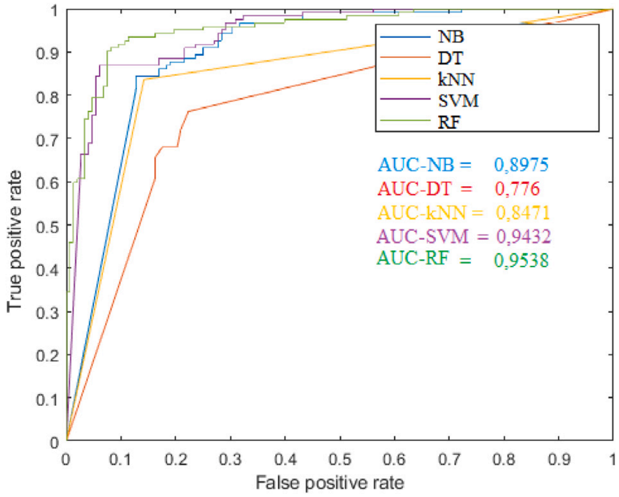

Fig. 2. ROC-curve of ResNet-101.

Table 5
Experimental results of ResNet-50.

	Accuracy (%)	Sensitivity (%)	Specificity (%)	F1-score (%)	Precision (%)
NB	83.333	84.426	82.432	82.072	79.845
SVM	96.296	95.082	97.297	95.868	96.667
kNN	90.741	83.607	96.622	89.083	95.327
DT	75.185	80.328	70.946	74.525	69.504
RF	93.704	91.803	95.27	92.946	94.118

5.1. Experimental results of ResNet-101

The deep features obtained with the ResNet-101 deep learning model were used in classification methods such as SVM, kNN, RF, DT and NB, and the results were compared. The results of the classification methods are reported in detail in Table 4. Accordingly, by looking at the specificity value, it is clear that SVM and RF methods are more successful than other methods in detecting negative images of COVID-19. In addition, since the specificity value is higher than the sensitivity value in all classification methods except NB, it can be said that negative image detection is more successful than positive image detection.

Fig. 2 shows the ResNet-101 ROC curve. As seen in the figure, the AUC value was 0.8975 in the NB method, 0.776 in the DT method, 0.8471 in the kNN method, 0.9432 in the SVM method and 0.9538 in the RF method. In the figure, it is seen that the classification method that best distinguishes the two classes is the RF method. Looking at the F1-score, accuracy value and ROC-curve, the Random Forest method obtained the highest values in all three success criteria.

5.2. Experimental results of ResNet-50

The deep features obtained with the ResNet-50 deep learning model were used in classification methods such as SVM, kNN, RF, DT and NB, and the results were examined. In Table 5, the results of the classification methods are given in detail. Accordingly, by looking at the specificity value, it is seen that SVM, RF and kNN methods are more successful in detecting negative images of COVID-19 than other methods. Since the SVM method showed very high success in detecting both negative and positive images, it achieved higher accuracy than all classification methods. Contrary to other methods, the sensitivity value of the NB and DT methods was higher than the specificity value. In other words, in these methods, COVID-19 positive images were detected better than negative images.

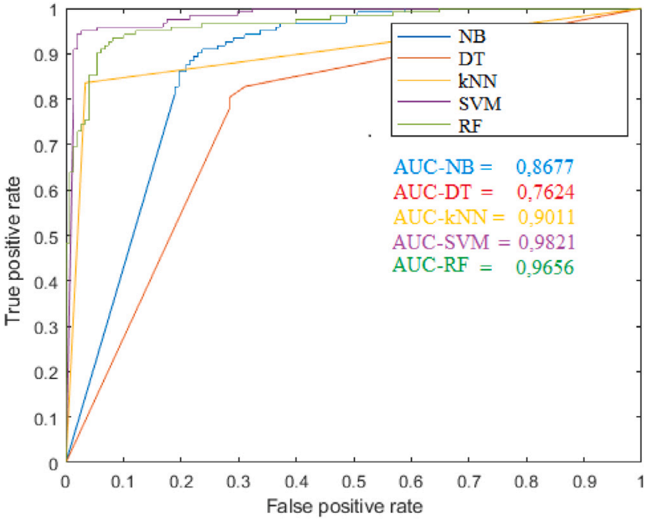

Fig. 3. ROC-curve of ResNet-50.

Table 6
Experimental results of AlexNet.

	Accuracy (%)	Sensitivity (%)	Specificity (%)	F1-score (%)	Precision (%)
NB	78.519	79.508	77.703	76.984	74.615
SVM	89.63	85.246	93.243	88.136	91.228
kNN	85.185	82.787	87.162	83.471	84.167
DT	74.074	81.967	67.568	74.074	67.568
RF	91.481	87.705	94.595	90.295	93.043

Fig. 3 shows the ResNet-50 ROC-curve. As seen in Fig. 3, the AUC value was 0.8677 in the NB method, 0.7624 in the DT method, 0.9011 in the kNN method, 0.9821 in the SVM method, and 0.9656 in the RF method. In the figure, it is seen that the classification method that best distinguishes the two classes is the SVM method. Looking at the F1-score, accuracy value and ROC curve, the SVM method showed the highest values in all three success criteria.

5.3. Experimental results of AlexNet

The deep features obtained with the AlexNet deep learning model were used in classification methods such as SVM, kNN, RF, DT and NB, and the results were compared with each other. The results of the classification methods are reported in detail in Table 6. Accordingly, by looking at the specificity value, it is clear that SVM and RF methods are more successful than other methods in detecting negative images of COVID-19.

Fig. 4 shows the AlexNet ROC-curve. As seen in Fig. 4, the AUC value was 0.8348 in the NB method, 0.7558 in the DT method, 0.8497 in the kNN method, 0.9384 in the SVM method, and 0.9492 in the RF method. In the figure, it is seen that the classification method that best distinguishes the two classes is the RF method. Looking at the F1-score, accuracy and ROC curve, the RF method showed the highest values in all three success criteria.

5.4. Experimental results of Vgg-19

The deep features obtained with the Vgg-19 deep learning model were used in classification methods such as SVM, kNN, RF, DT and NB, and the results were compared. The results of the classification methods are recorded in detail in Table 7. Accordingly, looking at the specificity value, it is seen that NB and RF methods are more successful in detecting negative images of COVID-19 than other methods. Considering the sensitivity value, SVM and RF methods obtained the most successful results in detecting positive images. Since the RF method achieved the highest results in both sensitivity and specificity, the accuracy value of the RF method was higher than the other methods. In addition, since the specificity value is higher than the sensitivity value in other classification methods except for SVM, it can be said that negative image detection is more successful than positive image detection.

Fig. 5 shows the AlexNet ROC-curve. As seen in Fig. 5, the AUC value was 0.9156 in the NB method, 0.7321 in the DT method, 0.8027 in the kNN method, 0.9119 in the SVM method, and 0.9362 in the RF method. In the figure, it is seen that the classification method that best distinguishes the two classes is the RF method. Looking at the F1-score, accuracy and ROC curve, the RF method showed the highest values in all three success criteria.

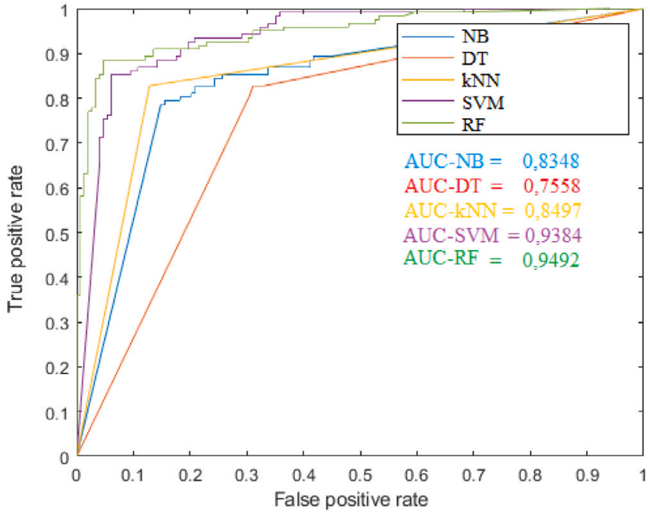

Fig. 4. ROC-curve of AlexNet.

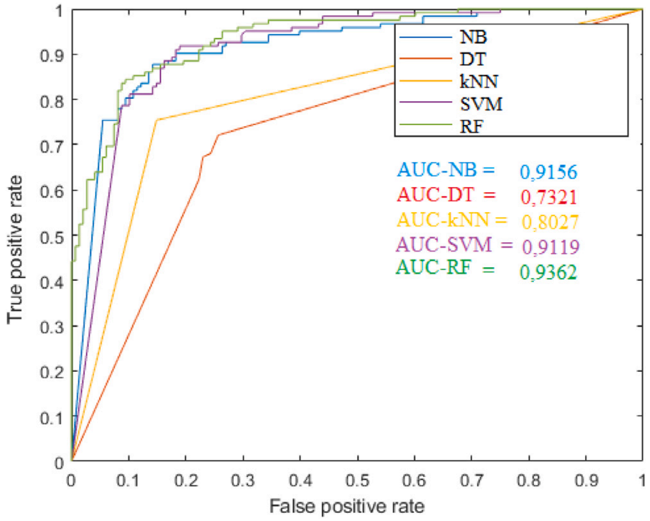

Fig. 5. ROC-curve of Vgg-19.

Table 7
Experimental results of Vgg-19.

	Accuracy (%)	Sensitivity (%)	Specificity (%)	F1-score (%)	Precision (%)
NB	85.926	81.967	89.189	84.034	86.207
SVM	85.926	88.525	83.784	85.039	81.818
kNN	80.741	75.41	85.135	77.966	80.702
DT	72.222	67.213	76.351	68.619	70.085
RF	87.778	86.066	89.189	86.42	86.777

5.5. Experimental results of Vgg-16

The results obtained by using deep features obtained by Vgg-16 deep learning model in classification methods such as SVM, kNN, RF, DT and NB were examined. The results of the classification methods are reported in detail in Table 8. Accordingly, by looking at the specificity and sensitivity value, it is clear that the SVM method is more successful than other methods in both detecting negative images of COVID-19 and detecting positive images. In addition, it can be said that negative image detection is more successful than positive image detection because the specificity value is higher than the sensitivity value in all classification methods.

Table 8
Experimental results of Vgg-16.

	Accuracy (%)	Sensitivity (%)	Specificity (%)	F1-score (%)	Precision (%)
NB	84.815	79.508	89.189	82.553	85.841
SVM	91.111	88.525	93.243	90	91.525
kNN	83.33	73.77	91.216	79.999	87.379
DT	77.037	74.459	79.054	74.59	74.59
RF	88.519	85.246	91.216	88.889	87.029

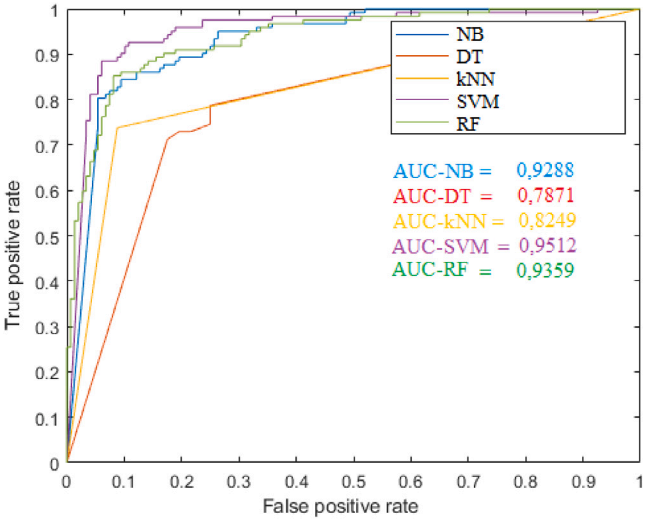

Fig. 6. ROC-curve of Vgg-16.

Table 9
Experimental results of GoogLeNet.

	Accuracy (%)	Sensitivity (%)	Specificity (%)	F1-score (%)	Precision (%)
NB	61.852	77.869	48.649	64.847	55.556
SVM	81.111	81.148	81.081	79.518	77.953
kNN	79.63	72.951	85.135	76.395	80.18
DT	71.852	72.131	71.622	69.841	67.692
RF	84.074	80.328	87.162	82.009	83.761

Fig. 6 shows the Vgg-16 ROC curve. As seen in Fig. 6, the AUC value was found to be 0.9288 in the NB method, 0.7871 in the DT method, 0.8249 in the kNN method, 0.9512 in the SVM method, and 0.9359 in the RF method. In the figure, it is seen that the classification method that best distinguishes the two classes is the SVM method. Looking at the F1-score, accuracy value and ROC curve, the SVM method showed the highest values in all three success criteria.

5.6. Experimental results of GoogLeNet

The deep features obtained with the GoogLeNet deep learning model were used in classification methods such as SVM, kNN, RF, DT and NB, and the results were compared. In Table 9, the results of the classification methods are given in detail. Accordingly, by looking at the specificity value, it is clear that kNN and RF methods are more successful than other methods in detecting negative images of COVID-19. The sensitivity value of the NB and DT methods was higher than the specificity value. In other words, in these methods, COVID-19 positive images were detected better than negative images.

Fig. 7 shows the GoogLeNet ROC curve. As seen in Fig. 7, the AUC value was 0.5994 in the NB method, 0.728 in the DT method, 0.7904 in the kNN method, 0.8405 in the SVM method, and 0.898 in the RF method. In the figure, it is seen that the classification method that best distinguishes the two classes is the RF method. Looking at the F1-score, accuracy and ROC curve, the RF method showed the highest values in all three success criteria.

5.7. Experimental results of SqueezeNet

The deep features obtained with the SqueezeNet deep learning model were used in classification methods such as SVM, kNN, RF, DT and NB, and the results were examined. The results of the classification methods are reported in detail in Table 10. Accordingly,

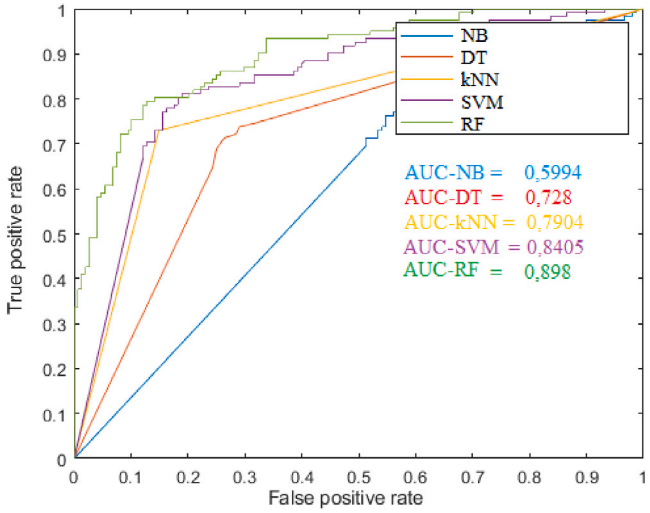

Fig. 7. ROC-curve of GoogLeNet.

Table 10
Experimental results of SqueezeNet.

	Accuracy (%)	Sensitivity (%)	Specificity (%)	F1-score (%)	Precision (%)
NB	80	67.213	90.541	75.229	85.417
SVM	89.63	87.705	91.216	88.43	89.167
kNN	78.889	63.934	91.216	73.239	85.714
DT	71.185	65.574	77.027	67.797	70.175
RF	88.889	80.328	95.946	86.726	94.231

Table 11
Experimental results of Xception.

	Accuracy (%)	Sensitivity (%)	Specificity (%)	F1-score (%)	Precision (%)
NB	83.704	83.607	83.784	82.258	80.952
SVM	89.63	86.885	91.892	88.333	89.831
kNN	87.037	85.246	88.514	85.597	85.95
DT	76.296	77.049	75.676	74.603	72.308
RF	88.889	91.803	86.486	88.189	84.848

it is clear that RF, SVM and kNN methods are successful in detecting negative images of COVID-19 by looking at the specificity value. Considering the sensitivity value, it was seen that the SVM method was more successful in detecting COVID-19 positive images than other methods. In addition, negative image detection was more successful than positive image detection in all classification methods.

Fig. 8 shows the SqueezeNet ROC curve. As seen in Fig. 8, the AUC value was 0.897 in the NB method, 0.7198 in the DT method, 0.7758 in the kNN method, 0.9415 in the SVM method, and 0.9516 in the RF method. In the figure, it is seen that the classification method that best distinguishes the two classes is the RF method. Looking at the F1-score and accuracy value, the SVM method showed the highest values, while the ROC curve showed the highest value in the RF method.

5.8. Experimental results of Xception

The deep features obtained with the Xception deep learning model were used in classification methods such as SVM, kNN, RF, DT and NB, and the results were compared. The results of the classification methods are reported in detail in Table 11. Accordingly, by looking at the specificity value, it is clear that the SVM method is more successful than other methods in detecting negative images of COVID-19. Considering the sensitivity value, it is seen that the RF method is more successful in detecting COVID-19 positive images than other methods. In addition, since the specificity value is higher than the sensitivity value in other classification methods except for DT and RF methods, it can be said that negative image detection is more successful than positive image detection.

The Xception ROC curve is given in Fig. 9. As seen in Fig. 9, the AUC value was 0.8946 in the NB method, 0.8001 in the DT method, 0.8688 in the kNN method, 0.9425 in the SVM method, and 0.9396 in the RF method. In the figure, it is seen that the classification method that best distinguishes the two classes is the SVM method. Looking at the F1-score, accuracy value and ROC curve, the SVM method showed the highest values in all three success criteria.

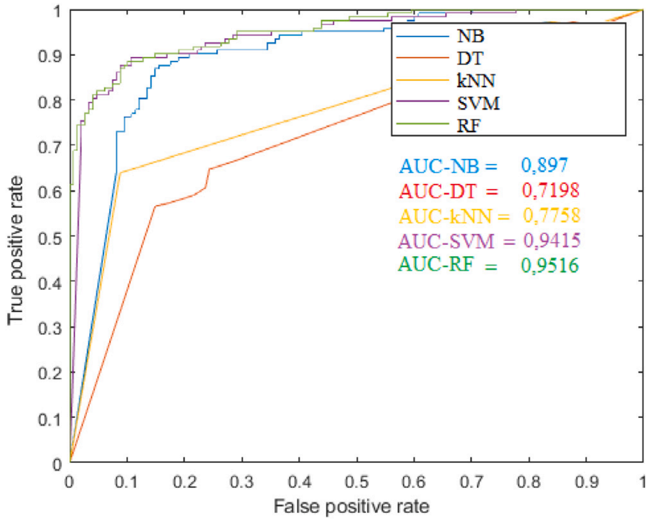

Fig. 8. ROC-curve of SqueezeNet.

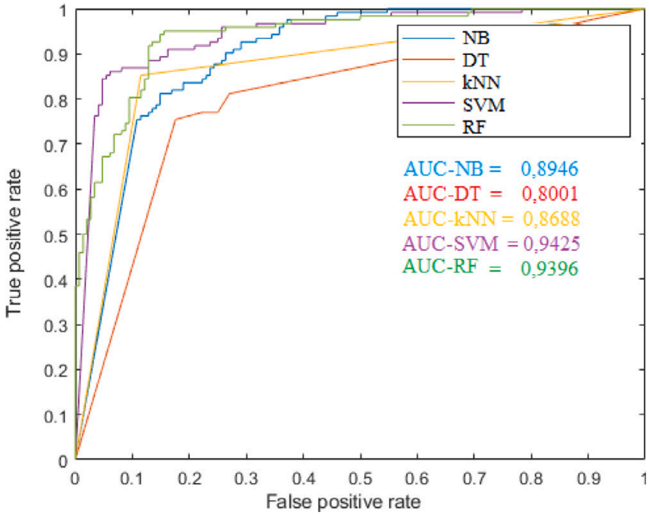

Fig. 9. ROC-curve of Xception.

In general, as seen in Table 12, the highest performance was obtained with the ResNet-50 and SVM methods. The accuracy value is 96.296%, the F1-score value is 95.858, and the AUC value is 0.9821. Looking at the runtime (RT), it is seen that ResNet-50 works faster than the other four models (ResNet-101, Xception, Vgg19 and Vgg-16). The lowest performance was obtained with GoogLeNet and NB method. The accuracy value was 61.852%, the F1-score value was 64.847, the AUC value was 0.5994, and the run time was 81.303 s Looking at the RT value in Table 12, the accuracy value of the fastest (22.58 s) AlexNet and kNN method is 85.249%. The slowest running (528.84 s) Vgg-19 and the accuracy value of the RF method is 87.778%. It is clear that AlexNet and SqueezeNet models are faster than other models. However, the accuracy values of SqueezeNet and GoogLeNet models are not high. The Vgg-19 is both slow and the accuracy value is lower than other models. In this case, it can be said that the best choice is ResNet-50. It can be said that the run time is reduced due to the small number of layers of the models and the 1 × 1 size filters. 1 × 1 filtering reduces size at depth. The pooling layer increases the number of channels while reducing the height and width of the input, that is, the depth is increased. As the depth increases, the run time increases. Except for the GoogLeNet model, the lowest performance was obtained with the DT method in other models.

As the number of epochs increases, the performance of the model increases significantly. Training can be terminated at these points, as performance will increase in very small units after a certain epoch. In this study, the training was terminated at the 6th epoch, as the success would increase very little after the 6th epoch in our trials. Hyper parameters were selected by trial and error method. By iteratively changing the hyperparameters one after the other, the success of the model was observed and the most suitable hyper parameter group was tried to be selected for the model. Fig. 10 shows the training and testing process of the ResNet-50

Table 12
Experimental results.

		Accuracy (%)	Sensitivity (%)	Specificity (%)	F1-score (%)	Precision (%)	AUC	RT (s)
ALEXNET	NB	78.519	79.508	77.703	76.984	79.508	0.8348	22.672
	SVM	89.63	85.246	93.243	88.136	85.246	0.9384	49.124
	kNN	85.185	82.787	87.162	83.471	82.787	0.8497	22.582
	DT	74.074	81.967	67.568	74.074	81.967	0.7558	23.181
	RF	91.481	87.705	94.595	90.295	87.705	0.9492	71.914
RESNET50	NB	83.333	84.426	82.432	82.072	84.426	0.8677	138.44
	SVM	96.296	95.082	97.297	95.868	95.082	0.9821	159.39
	kNN	90.741	83.607	96.622	89.083	83.607	0.9011	122.76
	DT	75.185	80.328	70.946	74.525	80.328	0.7624	133.78
	RF	93.704	91.803	95.27	92.946	91.803	0.9656	145.85
RESNET101	NB	82.963	83.607	82.432	81.6	83.607	0.8975	229.34
	SVM	90.37	86.066	93.919	88.983	86.066	0.9432	260.63
	kNN	84.815	83.607	85.811	83.266	83.607	0.8471	220.09
	DT	76.296	70.492	81.081	75.439	70.492	0.776	220.51
	RF	91.185	90.164	93.243	90.909	90.164	0.9538	247.92
VGG16	NB	84.815	79.508	89.189	82.553	79.508	0.9288	362.52
	SVM	91.111	88.525	93.243	90	88.525	0.9512	414.68
	kNN	83.33	73.77	91.216	79.999	73.77	0.8249	419.68
	DT	77.037	74.459	79.054	74.59	74.459	0.7871	380.75
	RF	88.519	85.246	91.216	88.889	85.246	0.9359	387.55
VGG19	NB	85.926	81.967	89.189	84.034	81.967	0.9156	429.28
	SVM	85.926	88.525	83.784	85.039	88.525	0.9119	518.31
	kNN	80.741	75.41	85.135	77.966	75.41	0.8027	482.86
	DT	72.222	67.213	76.351	68.619	67.213	0.7321	451.48
	RF	87.778	86.066	89.189	86.42	86.066	0.9362	528.84
SQUEEZENET	NB	80	67.213	90.541	75.229	67.213	0.897	41.049
	SVM	89.63	87.705	91.216	88.43	87.705	0.9415	56.241
	kNN	78.889	63.934	91.216	73.239	63.934	0.7758	56.330
	DT	71.185	65.574	77.027	67.797	65.574	0.7198	42.535
	RF	88.889	80.328	95.946	86.726	80.328	0.9516	51.164
XCEPTION	NB	83.704	83.607	83.784	82.258	83.607	0.8946	339.08
	SVM	89.63	86.885	91.892	88.333	86.885	0.9425	416.96
	kNN	87.037	85.246	88.514	85.597	85.246	0.8688	337.14
	DT	76.296	77.049	75.676	74.603	77.049	0.8001	356.01
	RF	88.889	91.803	86.486	88.189	91.803	0.9396	373.29
GOOGLENET	NB	61.852	77.869	48.649	64.847	77.869	0.5994	81.303
	SVM	81.111	81.148	81.081	79.518	81.148	0.8405	86.217
	kNN	79.63	72.951	85.135	76.395	72.951	0.7904	83.822
	DT	71.852	72.131	71.622	69.841	72.131	0.728	232.09
	RF	84.074	80.328	87.162	82.009	80.328	0.898	98.620

Table 13
Classification experimental results with ResNet-50.

Accuracy (%)	Sensitivity (%)	Specificity (%)	F1-score (%)	Precision (%)	RT (s)
92.59	94.262	91.216	91.999	89.844	7458

model. As can be seen in the figure, training was conducted for a maximum of 6 epochs. An epoch is a complete training cycle on the entire training dataset. The fc layer, softmax layer and output layer are changed and transferred to the new classification task. In these layers, the learning rate is set to 20 and learning is accelerated in the last layers. The initial learning rate in the transfer layers was determined as 0.0001. Thus, learning is slowed down in these layers. The verification frequency is set to 10 iterations.

As can be seen in Table 13, using deep learning models together with classification methods, instead of using only deep learning model (ResNet-50), reduces the run time. While classifying with ResNet-50, 7458 s run time is calculated. When ResNet-50 and SVM methods classify together, 159.39 s run time is calculated. A deep learning algorithm takes a long time to train because there are so many parameters. Traditional machine learning takes relatively little time, ranging from a few minutes to several hours. Traditional machine learning systems can be set up and run quickly, but these systems require structured data. In complex problems, the power of its results may be limited. Deep learning systems take more time to set up, but using deep learning for complex computations (unstructured data) will strengthen results. Therefore, in this study, it can be said that by creating a hybrid model, extracting features with deep learning models and using them in traditional machine learning methods shortens the time and strengthens the results. ResNet-50 alone detected the COVID-19 images with 94.262% accuracy, but when ResNet-50 and SVM method were used together, it detected the COVID-19 images with 95.082% accuracy.

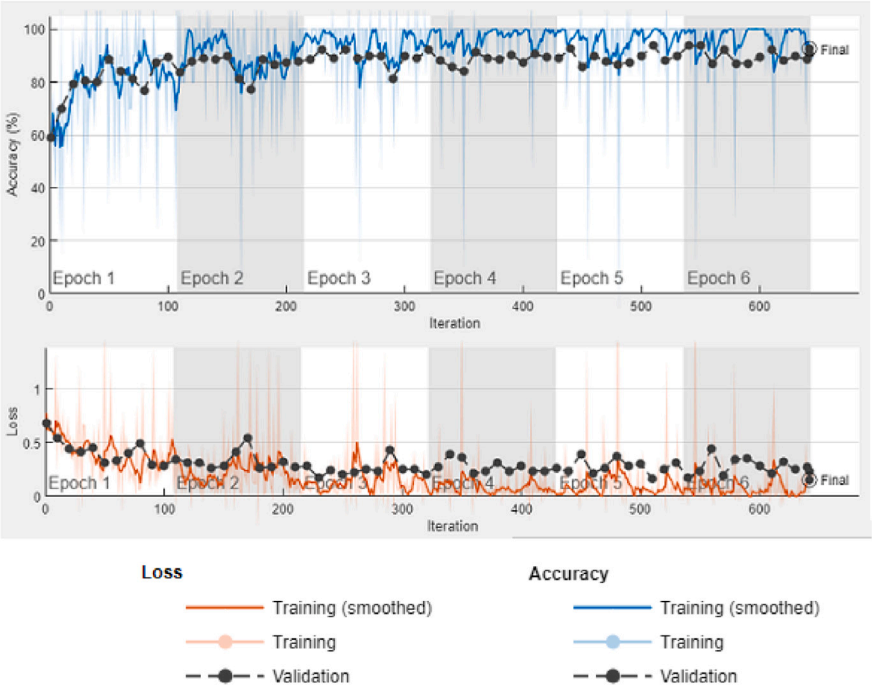

Fig. 10. The Training and Testing Process of ResNet-50.

6. Discussion

We combined classification methods (traditional machine learning) and capabilities extracted from deep learning models to increase the accuracy value in COVID-19 detection. In this review, we obtained deep features for COVID-19 detection using 8 powerful deep learning models on CT images. Comparative analysis was performed using 5 different classification methods for each deep feature. On evaluating We concluded that using the deep learning model and classification method together provides greater accuracy in the evaluation. Using deep learning models together with classification methods, instead of using only deep learning model (ResNet-50), reduces the run time. While classifying with ResNet-50, 7458 s run time is calculated. When ResNet-50 and SVM methods classify together, 159.39 s run time is calculated. ResNet-50 alone detected the COVID-19 positive images with 94.262% accuracy, but when ResNet-50 and SVM method were used together, it detected the COVID-19 images with 95.082% accuracy.

Alternatively, using ResNet-50 and SVM together, it achieved 96.296% COVID-19 detection accuracy for the dataset. The purpose of this study is to reduce the duration and amount of transmission of the disease by shortening the diagnosis time by taking CT images from patients who were admitted to the hospital due to the suspicion of COVID-19. In addition, it is aimed to provide a decision support system to radiologists in the detection of COVID-19.

The COVID-19 pandemic is a new disease problem, and there is no definite and sufficient data in the literature on this subject. All studies except (Wu et al., 2021) study in the literature could not reach the COVID-19 patient image as in this study. The data set used in this study consists of real images obtained from the hospital environment. In addition, images thought to be positive for COVID-19 in this study were confirmed by a radiologist. The working times of the methods used in most of the studies in the literature are not mentioned. In this study, methods were evaluated according to accuracy, sensitivity, specificity, precision, F1-score, AUC and run time. Compared to the studies in the literature, a broad evaluation was made in this study using various classification methods together with deep learning models.

The performance evaluation of the proposed model compared to studies using CT images in the literature is given in Table 14. Since it is more important to accurately detect COVID-19 positive cases, sensitivity values are taken into account in Table 14. The sensitivity value of the proposed model (95.082%) achieved higher performance than all models based on CT images in the studies in the literature.

Pham (2020) examined transfer learning (pretrained convolutional neural networks) methods in his study. In our study, various hybrid models were created by using the deep features obtained from different deep learning methods in different machine learning methods for the detection of COVID-19. The 1345 CT images in the dataset used were obtained from the hospital environment. It consists of various CT images (COVID-19 early stage or later stages, different ages and different genders). The performances of the new hybrid models created were examined. In the results of the experiment, it was seen that more successful results were obtained by using the hybrid model created by integrating deep learning and machine learning methods, as in our study, instead of using deep learning models alone as in the Pham (2020) study.

Table 14
Comparison of the proposed model with studies using CT images.

Study	Method	Number of images	Data source	Performance
Wu et al. (2020)	COVNet	4356	Multiple hospital environment	90% sensitivity
Zheng et al. (2020)	DeCOVNet	630	Multiple hospital environment	90.7% sensitivity
Singh et al. (2020)	MODE-CNN	150	Open source database	90.7% sensitivity
Pham (2020)	DenseNet-201	746	Open source database	96.2% accuracy
Wu et al. (2021)	COVIDAL	962	Open source database	86.6% accuracy
Rohila et al. (2021)	ReCOV-11	1110	MosMedData	94.9% accuracy
Wang, Kang et al. (2021)	Modified Inception V3	1065	Multiple hospital environment	85.2% accuracy
Li et al. (2021)	CheXNet	746	Open source database	87% accuracy
This study	ResNet-50 - SVM	1345	Single hospital environment	96.96% accuracy 95.082% sensitivity

The COVID-19 pandemic is a new disease problem, and there is no definite and sufficient data in the literature on this subject. All studies except (Wu et al., 2020) study in the literature could not reach the COVID-19 patient image as in this study. The data set collected consists of real images obtained from the single hospital environment. Images thought to be positive for COVID-19 in this study were confirmed by a radiologist.

Most of the datasets used were created from open-source databases. The data set in some studies in the literature (Mahmud et al., 2020; Oh et al., 2020; Ucar & Korkmaz, 2020) contains a large number of images, but the number of images of COVID-19 cases is low. In addition, some studies (Panwar et al., 2020; Singh et al., 2020) have datasets containing a small number of images. In both cases, a limited number of data leads to underfitting or overfitting, reducing the performance of deep learning models (Islam et al., 2021; Nayak et al., 2021). In some of the studies in the literature (Mahmud et al., 2020; Oh et al., 2020; Ucar & Korkmaz, 2020) data imbalance is observed. Data imbalance also increases bias to a class (Islam et al., 2021). The run times of the methods used in most of the studies in the literature are not mentioned.

7. Conclusion

Since the patient is not quarantined during the conclusion of the Polymerase Chain Reaction (PCR) test used in the diagnosis of COVID-19, the disease continues to spread. In this study, it was aimed to reduce the duration and amount of transmission of the disease by shortening the diagnosis time of COVID-19 patients with the use of Computed Tomography (CT). In addition, it is aimed to provide a decision support system to radiologists in the diagnosis of COVID-19.

In this study, a new high-performance COVID-19 detection system was developed with the hybrid model created from the combination of ResNet-50 and SVM. In our proposed hybrid model framework, there are 2 modules including feature extraction with deep learning method and classification with traditional machine learning method. The data set used in this study consists of real images obtained from a single hospital environment. Images thought to be positive for COVID-19 were confirmed by a radiologist. According to the experimental results, the proposed hybrid model (ResNet-50+SVM) achieved higher accuracy than the classical ResNet-50 method.

In the future, images from more than one hospital can be collected and better results can be obtained. Different lung diseases can be added in future studies.

References

Afify, H. M., & Zanaty, M. S. (2021). Computational predictions for protein sequences of COVID-19 virus via machine learning algorithms. *Medical & Biological Engineering & Computing*, *59*(9), 1723–1734.

Ardakani, A. A., Kanafi, A. R., Acharya, U. R., Khadem, N., & Mohammadi, A. (2020). Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks. *Computers in Biology and Medicine*, *121*, Article 103795.

Bhole, V., & Kumar, A. (2020). Mango quality grading using deep learning technique: Perspectives from agriculture and food industry. In *Proceedings of the 21st annual conference on information technology education* (pp. 180–186).

Bozkurt, F. (2021). A comparative study on classifying human activities using classical machine and deep learning methods. *Arabian Journal for Science and Engineering*, 1–15.

Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, *2*(01), 20–28.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251–1258).

Fang, Y., Zhang, H., Xie, J., Lin, M., Ying, L., Pang, P., et al. (2020). Sensitivity of chest CT for COVID-19: comparison to RT-PCR. *Radiology*, *296*(2), E115–E117.

Firouzi, F., Farahani, B., Daneshmand, M., Grise, K., Song, J. S., Saracco, R., et al. (2021). Harnessing the power of smart and connected health to tackle COVID-19: Iot, AI, robotics, and blockchain for a better world. *IEEE Internet of Things Journal*, *8*(16).

Fleuren, L. M., Dam, T. A., Tonutti, M., de Bruin, D. P., Lalisang, R. C., Gommers, D., et al. (2021). The dutch data warehouse, a multicenter and full-admission electronic health records database for critically ill COVID-19 patients. *Critical Care*, *25*(1), 1–12.

Fleuren, L. M., Tonutti, M., de Bruin, D. P., Lalisang, R. C., Dam, T. A., Gommers, D., et al. (2021). Risk factors for adverse outcomes during mechanical ventilation of 1152 COVID-19 patients: a multicenter machine learning study with highly granular data from the Dutch Data Warehouse. *Intensive Care Medicine Experimental*, *9*(1), 1–15.

Garcia, P. D. W., Aguirre-Bermeo, H., Buehler, P. K., Alfaro-Farias, M., Yuen, B., David, S., et al. (2021). Implications of early respiratory support strategies on disease progression in critical COVID-19: a matched subanalysis of the prospective RISC-19-ICU cohort. *Critical Care*, *25*(1), 1–12.

Hu, L.-Y., Huang, M.-W., Ke, S.-W., & Tsai, C.-F. (2016). The distance function effect on k-nearest neighbor classification for medical datasets. *SpringerPlus*, *5*(1), 1–9.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-Level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv preprint arXiv:1602.07360.

Islam, M. M., Karray, F., Alhajj, R., & Zeng, J. (2021). A review on deep learning techniques for the diagnosis of novel coronavirus (covid-19). *IEEE Access*, *9*, 30551–30572.

Jahanbani Fard, M. (2015). BayesIan approach for early stage event prediction in survival data. URL: https://digitalcommons.wayne.edu/oa_theses/455.

Kang, Z., Li, X., & Zhou, S. (2020). Recommendation of low-dose CT in the detection and management of COVID-2019.

Li, C., Yang, Y., Liang, H., & Wu, B. (2021). Transfer learning for establishment of recognition of COVID-19 on CT imaging using small-sized training datasets. *Knowledge-Based Systems*, *218*, Article 106849.

Mahmud, T., Rahman, M. A., & Fattah, S. A. (2020). 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. *Computers in Biology and Medicine*, *122*, Article 103869.

Nayak, J., Naik, B., Dinesh, P., Vakula, K., Dash, P. B., & Pelusi, D. (2021). Significance of deep learning for Covid-19: state-of-the-art review. *Research on Biomedical Engineering*, 1–24.

Oh, Y., Park, S., & Ye, J. C. (2020). Deep learning covid-19 features on cxr using limited training data sets. *IEEE Transactions on Medical Imaging*, *39*(8), 2688–2700.

Panahi, A. H., Rafiei, A., & Rezaee, A. (2021). FCOD: FAst COVID-19 detector based on deep learning techniques. *Informatics in Medicine Unlocked*, *22*, Article 100506.

Panwar, H., Gupta, P., Siddiqui, M. K., Morales-Menendez, R., & Singh, V. (2020). Application of deep learning for fast detection of COVID-19 in X-Rays using ncovnet. *Chaos, Solitons & Fractals*, *138*, Article 109944.

Pham, T. D. (2020). A comprehensive study on classification of COVID-19 on computed tomography with pretrained convolutional neural networks. *Scientific Reports*, *10*(1), 1–8.

Rohila, V. S., Gupta, N., Kaul, A., & Sharma, D. K. (2021). Deep learning assisted COVID-19 detection using full CT-scans. *Internet of Things*, *14*, Article 100377.

Romeo, L., & Frontoni, E. (2022). A unified hierarchical xgboost model for classifying priorities for COVID-19 vaccination campaign. *Pattern Recognition*, *121*, Article 108197.

Shui-Hua, W., Khan, M. A., Govindaraj, V., Fernandes, S. L., Zhu, Z., & Yu-Dong, Z. (2022). Deep rank-based average pooling network for COVID-19 recognition. *Computers, Materials, & Continua*, 2797–2813.

Singh, D., Kumar, V., Kaur, M., et al. (2020). Classification of COVID-19 patients from chest CT images using multi-objective differential evolution–based convolutional neural networks. *European Journal of Clinical Microbiology & Infectious Diseases*, *39*(7), 1379–1389.

Ucar, F., & Korkmaz, D. (2020). COVIDiagnosis-Net: DEep Bayes-SqueezeNet based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images. *Medical Hypotheses*, *140*, Article 109761.

Wang, S., Kang, B., Ma, J., Zeng, X., Xiao, M., Guo, J., et al. (2021). A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19). *European Radiology*, *31*(8), 6096–6104.

Wang, S.-H., Zhang, X., & Zhang, Y.-D. (2021). DSSAE: DEep stacked sparse autoencoder analytical model for COVID-19 diagnosis by fractional Fourier entropy. *ACM Transactions on Management Information System (TMIS)*, *13*(1), 1–20.

Wu, X., Chen, C., Zhong, M., Wang, J., & Shi, J. (2021). COVID-AL: THe diagnosis of COVID-19 with deep active learning. *Medical Image Analysis*, *68*, Article 101913.

Wu, X., Hui, H., Niu, M., Li, L., Wang, L., He, B., et al. (2020). Deep learning-based multi-view fusion model for screening 2019 novel coronavirus pneumonia: A multicentre study. *European Journal of Radiology*, *128*, Article 109041.

Yang, N., Liu, S., Liu, J., & Li, C. (2021). Assessment of equipment operation state with improved random forest. *International Journal of Rotating Machinery*, *2021*.

Yuvaraj, N., Chang, V., Gobinathan, B., Pinagapani, A., Kannan, S., Dhiman, G., et al. (2021). Automatic detection of cyberbullying using multi-feature based artificial intelligence with deep decision tree classification. *Computers and Electrical Engineering*, *92*, Article 107186.

Zhang, Y.-D., Zhang, Z., Zhang, X., & Wang, S.-H. (2021). MIDCAN: A Multiple input deep convolutional attention network for Covid-19 diagnosis based on chest CT and chest X-ray. *Pattern Recognition Letters*, *150*, 8–16.

Zheng, C., Deng, X., Fu, Q., Zhou, Q., Feng, J., Ma, H., et al. (2020). Deep learning-based detection for COVID-19 from chest CT using weak label. *MedRxiv*, http://dx.doi.org/10.1101/2020.03.12.20027185.

Zhou, Z.-H. (2021). Support vector machine. In *Machine learning* (pp. 129–153). Springer.