Neural Computing and Applications (2022) 34:8253–8274
https://doi.org/10.1007/s00521-022-06918-x

ORIGINAL ARTICLE

VGGCOV19-NET: automatic detection of COVID-19 cases from X-ray images using modified VGG19 CNN architecture and YOLO algorithm

Abdulkadir Karacı[1]

Received: 15 May 2021 / Accepted: 4 January 2022 / Published online: 24 January 2022

Abstract

X-ray images are an easily accessible, fast, and inexpensive method of diagnosing COVID-19, widely used in health centers around the world. In places where there is a shortage of specialist doctors and radiologists, there is need for a system that can direct patients to advanced health centers by pre-diagnosing COVID-19 from X-ray images. Also, smart computer-aided systems that automatically detect COVID-19 positive cases will support daily clinical applications. The study aimed to classify COVID-19 via X-ray images in high precision ratios with pre-trained VGG19 deep CNN architecture and the YOLOv3 detection algorithm. For this purpose, VGG19, VGGCOV19-NET models, and the original Cascade models were created by feeding these models with the YOLOv3 algorithm. Cascade models are the original models fed with the lung zone X-ray images detected with the YOLOv3 algorithm. Model performances were evaluated using fivefold cross-validation according to recall, specificity, precision, f1-score, confusion matrix, and ROC analysis performance metrics. While the accuracy of the Cascade VGGCOV19-NET model was 99.84% for the binary class (COVID vs. no-findings) data set, it was 97.16% for the three-class (COVID vs. no-findings vs. pneumonia) data set. The Cascade VGGCOV19-NET model has a higher classification performance than VGG19, Cascade VGG19, VGGCOV19-NET and previous studies. Feeding the CNN models with the YOLOv3 detection algorithm decreases the training test time while increasing the classification performance. The results indicate that the proposed Cascade VGGCOV19-NET architecture was highly successful in detecting COVID-19. Therefore, this study contributes to the literature in terms of both YOLO-aided deep architecture and classification success.

Keywords Coronavirus · COVID-19 · Chest X-ray images · Deep CNN · Transfer learning · YOLO

1 Introduction

A new pneumonia epidemic called acute respiratory syndrome coronavirus 2 (COVID-19) started in Wuhan in the Hubei province of China in December 2019 [1–3]. COVID-19 rapidly expanded from a single city to the whole country within 30 days. This sudden increase in cases caused the collapse of the health system in China [4, 5]. COVID-19 is transmitted from person to person through direct contact and by small droplets emitted by infected people. Similar to the flu, a COVID-19 patient can develop a variety of symptoms and signs of infection, such as fever, cough, and respiratory illness. In more severe cases, the infection can cause breathing difficulties, multiple organ failure, and a rapidly progressive fatal pneumonia in 2–8% of those infected. Pneumonia can be detected from chest X-ray images [6–8].

Chest radiological imaging such as X-ray and computerized tomography (CT) plays a vital role in the early diagnosis and treatment of this disease [1, 9]. X-rays can detect several characteristic signs associated with COVID-19 in the lung [9]. Early results show that patients suggestive of COVID-19 have abnormalities on chest X-rays [10]. The real-time reverse-transcription polymerase chain reaction (RT-PCR) method is also frequently used to diagnose COVID-19. However, the sensitivity of this method is around 60–70%. Therefore, it is important to diagnose patients by examining X-ray images [1, 9]. In

✉ Abdulkadir Karacı
akaraci@kastamonu.edu.tr

[1] Faculty of Engineering and Architecture, Computer Engineering, Kastamonu University, 37200 Kastamonu, Turkey

addition, a detection system based on X-ray images has many advantages. It is fast, can analyze multiple cases at the same time, and has more usability. In addition, X-ray equipment is available in every hospital in the modern healthcare system. This makes the radiography-based approach convenient and easily accessible [11]. CT scans and X-rays are widely used for the detection of COVID-19 in countries in which there are few test kits. Also, researchers specify that the combination of clinical image properties with laboratory results may be helpful in the early diagnosis of COVID-19 [9, 12, 13]. There are also studies reporting that some changes have been encountered in the chest X-rays and CT images of asymptomatic children and the elderly in the early periods of the disease [14]. These changes can be detected by deep convolutional neural networks (CNN).

Deep CNN outperformed all other known methods in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) 2012 [15]. Deep CNNs are among the strong deep learning architectures that are widely and instinctively applied in many practical applications such as pattern recognition and image classification [16]. The greatest advantage of CNN for image classification is the end-to-end training of the whole system from raw pixels to the final categories. This advantage decreases the need to design a manually convenient feature extractor [15]. Researchers are focusing on deep learning techniques to determine specific properties of COVID-19 patients from the chest radiography images [17, 18]. Recently, deep learning has been highly successful in various visual tasks including medical image analysis. Deep learning has revolutionized the automatic diagnosis and management of the disease by correctly analyzing, defining and classifying the patterns in medical images [11]. In addition, transfer learning is widely used in CNN architectures. Transfer learning is used to improve the learner by transferring the information in a related area. It aims to apply the data extracted from one or more tasks to the target task. Together with the growth of deep learning, transfer learning has become an inseparable part of many applications especially in medical imaging [19, 20]. There are many pre-trained deep CNN architectures (VGG16, VGG19, MobileNet, Inception, Xception, ResNet, DarkNet, AlexNet, etc.) in which the transfer learning approach can be applied.

The Visual Geometry Group Network (VGG) was developed based on the convolutional neural network architecture at the Oxford Robotics Institute [16] and was introduced by Simonyan and Zisserman [21]. VGGNet has shown particularly good performance in the ImageNet data cluster. This network has been trained on more than 1 million images in 1000 classes with more than 370,000 iterations to calibrate 138 million weight parameters. In particular, VGG19 won first place in a classification and localization competition in the Large-Scale Visual Recognition Competition, a global image recognition competition, in 2014 [21, 22]. The VGG is a series of transfer learning networks that includes VGG11, VGG13, VGG16, and VGG19. The common feature of these network structures is that several convolution layer modules are connected to three full connection layers [23]. VGG19 consists of five building blocks. The first and second building blocks consist of two convolutional layers and one pooling layer. The third and fourth blocks have four convolutional layers and one pooling layer. The final block consists of four convolutional layers. 3 × 3 small filters are also used [24].

The detection of the zone that should be the area of focus and supplying it to these CNN architectures as input instead of raw images may increase the classification performance. Detection algorithms are used for this process. One of these algorithms is YOLO (You Only Look Once). The YOLO architecture consists of 27 CNN layers with 24 convolution layers, followed by two fully connected layers. YOLOv3 is a deep single-stage CNN developed from YOLO and YOLOv2 for object detection. The YOLOv3 algorithm is one of the best object detection methods and it provides perfect improvements in terms of speed. The YOLO-V3 has high detection accuracy and speed. It also performs well in detecting small targets. This algorithm reframes object detection as a single regression problem, from image pixels to bounding box coordinates and class probabilities. It works significantly faster than other detection methods. It uses a different architecture called Darknet-53 for feature extraction. DarkNet-53 uses 3 × 3 and 1 × 1 convolutional layers and has 53 layers [25–28].

This study aims to develop a new state-of-the-art VGG19 (Visual Geometry Group Network) architecture-based improved model to diagnose COVID-19 automatically from chest X-rays, to reveal a new cascade model by combining this developed model with the YOLOv3 [25] detection algorithm, and to evaluate the efficiency of these models. For this purpose, the fully connected (FC) layer of the pre-trained VGG19 CNN architecture was re-arranged and attempts were made to diagnose COVID-19 from binary (COVID-19 vs. normal) and three-class (COVID-19 vs. normal vs. pneumonia) data sets. The modified VGG19 model was also combined with the YOLOv3 algorithm to detect the lung zone and the classification process was conducted with this newly formed cascade model. The performances of these models were compared to each other and to models in previous studies.

The main contributions of this study can be summarized as follows.

- We propose the VGG19-based improved VGGCOV19-NET (Visual Geometry Group Network COVID19-

NET) and Cascade VGGCOV19-NET models to assist in the automatic diagnosis of COVID-19 patients.

- Via the VGGCOV19-NET model, we prove that the classification performances of the pre-trained CNN architectures can be enhanced as a result of modification.
- We propose a new Cascade VGGCOV19-NET model by feeding the VGGCOV19-NET model with the lung zone X-ray images detected with the YOLOv3 algorithm.
- The new Cascade VGGCOV19-NET model significantly decreases the training and test time while significantly increasing the classification performance.
- The COVID-19 classification performance of Cascade VGGCOV19-NET is generally higher than that in previous studies.
- COVID-19 is rapidly diagnosed without manual feature extraction and selection methods.

The manuscript is organized as follows: Previous studies are summarized in Part 2. The structures, parameters, and training of Cascade VGGCOV19-NET and VGGCOV19-NET model and the data sets and metrics used in the study are explained in Part 3. Results and classification performances are presented in Part 4. The classification performances of models are compared to that of previous studies and the results are discussed in Part 5. Finally, in Part 6, the conclusion is given.

2 Literature review

There are some studies using radiology images and machine learning to diagnose COVID-19 in the literature. Hemdan et al. [16] tried different CNN classifiers to diagnose COVID-19 from X-ray images. They attained the highest classification performance from VGG19 and DenseNet201 classifiers with 90% accuracy. Wang et al. [29] classified the normal, pneumonia, and COVID-19 classes with 93.3% accuracy with a CNN model they called COVID-Net. Aslan et al. [30] classified the normal, pneumonia, and COVID-19 classes with 98.70% accuracy using hybrid CNN model. Hira et al. [31] classified the normal, pneumonia and COVID-19 classes with 97.55% accuracy using ResNeXt-50. Khan et al. [11] classified three classes (COVID-19, pneumonia, and normal) with 95% accuracy and binary class (COVID-19 and normal) with 99% accuracy using CNN. Medhi et al. [32] diagnosed COVID-19 with 93% accuracy with their CNN model. Ozturk et al. [9] classified three classes (COVID-19, pneumonia, and normal) with 87.02% accuracy and the binary class with 98.08% accuracy with the CNN called DarkCovidNet. Harit et al. [33] classified COVID-19 with 94% accuracy using ResNet. Ahammed et al. [34] classified three classes (COVID-19, pneumonia, and normal) with 18 different classifiers. They attained the highest accuracy value (94%) in their CNN model.

Apostolopoulos and Mpesiana [35] classified three classes and binary classes using the transfer learning approach with five different state-of-the-art pre-trained CNNs. They attained the highest accuracy in the VGG19 model. These accuracy values are, respectively, 98.75% and 93.48% for the binary class and three-class classification. Similarly, Narin et al. [36] classified the binary class using the transfer learning approach with three different state-of-the-art pre-trained CNNs. They attained the highest accuracy value (98%) in the ResNet model. Benbrahim et al. [37] classified COVID-19 with 99% accuracy with the InceptionV3 model and transfer learning approach.

Butt et al. [38] classified three classes with 86.7% accuracy on chest CT images with the ResNet model. Ying et al. [39] classified COVID-19 with 94% accuracy with the DRE-Net CNN model from chest CT images. Zheng et al. [40] diagnosed COVID-19 with 90.1% accuracy from chest CT images with the 3D deep CNN model. Harmon et al. [7] detected COVID-19 with 90.8% accuracy using the Densnet-121 CNN architecture on a total of 2724 CT images, 1029 of which were COVID-19 positive. Waheed et al. [10] stated that it is difficult to collect X-ray images because the epidemic is new, and they proposed a method to produce synthetic X-ray images. They suggested that when synthetic X-ray images produced by this method are added to the data set, the model performance increases.

3 Method and materials

In this part, the structure of the VGGCOV19-NET, Cascade VGGCOV19-NET, VGG19, and Cascade VGG19 models, the data set used in the formation of the models, and the performance metrics used in the evaluation of the models are explained.

3.1 X-ray image data set

The data set used in this study was attained from the X-ray images used by Ozturk et al. [9] in their study. They have published this data set as an open resource at https://github.com/muhammedtalo/COVID-19. The data set consists of 125 COVID-19, 500 pneumonia, and 500 no-findings X-ray images. They used two different resources in the formation of the data set: COVID-19 X-ray images from the database attained from various open access resources by Cohen [41] and the ChestX-ray8 database provided by Wang et al. [42] for normal and pneumonia X-ray images. Forty-three of the patients in the COVID-19 data set are women and 82 are men. Metadata information for this data set are not provided for all patients. The average age of the 26 COVID-19 positive patients is approximately 55 years. It is a data set convenient for both binary and multi-class classifications. Figure 1 shows the X-ray images of some cases with COVID-19, pneumonia, and no-findings taken from the data set.

3.2 The proposed VGGCOV19-NET and cascade VGGCOV19-NET models

Deep learning or deep structured learning represents a class of improved machine learning techniques [37]. The occurrence of deep learning technology has been a revolution in artificial intelligence. The word "deep" expresses the increase in the dimension of this network in addition to the number of layers [9]. CNN is the deep learning approach commonly used for feature extraction and data classification in various areas. CNN has many layers, the input layer, the convolutional layer, the pooling layer, the fully connected layer, and the output layer [43, 44]. The most efficient way to use deep CNNs on small data sets is transfer learning [45]. The data gained by the pre-trained model on a large data set is transferred on the model to be trained with the transfer learning method commonly used in deep learning. The greatest advantage of using the transfer learning method is that it allows the training of fewer data clusters and incurs lower calculation costs [36]. The Visual Geometry Group Network (VGG) was developed based on the convolutional neural network architecture at the Oxford Robotics Institute [16]. VGGNet has shown particularly good performance in the ImageNet data cluster. VGG19 consists of five building blocks. The first and second building blocks consist of two convolutional layers and one pooling layer. The third and fourth blocks have four convolutional layers and one pooling layer. The final block consists of four convolutional layers. 3 × 3 small filters are also used [24, 46].

In this study, the suggested VGGCOV19-NET model is based on the modified VGG19 [21] to diagnose COVID-19. The Cascade VGGCOV19-NET model is a hybrid model in which the VGGCOV19-NET model and the YOLOv3 detection algorithm are used together. The architectures of the VGGCOV19-NET and Cascade VGGCOV19-NET models are shown in Fig. 2. The first five blocks of VGG19 are out of the top layer in VGGCOV19-NET, and they have been transferred to the new model with their weights. The top layer has been made compatible to the data with 224 × 224 × 3 dimension and the FC layer has been reformed. The FC layer has one Flatten and three dense layers in the raw state of the VGG19 model. The first two dense layers consist of 4096 neurons. The final layer is the output layer that consists of 1000 neurons. The modified FC layer of the VGGCOV19-NET model has one flatten and four Dense layers. The neuron numbers in the first three dense layers are, respectively, 128, 256 and 512. The Relu activation function was used in these layers. The final dense layer is the output layer whose activation function is Softmax. The neuron number in this layer is two (COVID-19 and no-findings) for binary classification and three (COVID-19, no-findings, and pneumonia) for multi-class classification. The FC layer is given in Fig. 3 with a more detailed image. As seen in this figure, the flatten layer output at the front of the FC layer consists of 25,088 features. As a result, the input layer of the FC layer has 25,088 neurons. There are 150,528 data for each X-ray image. These data are filtered, and 25,088 features are extracted in convolution and pooling layers that are connected from the front to the flatten layer. The dense layer number, neuron

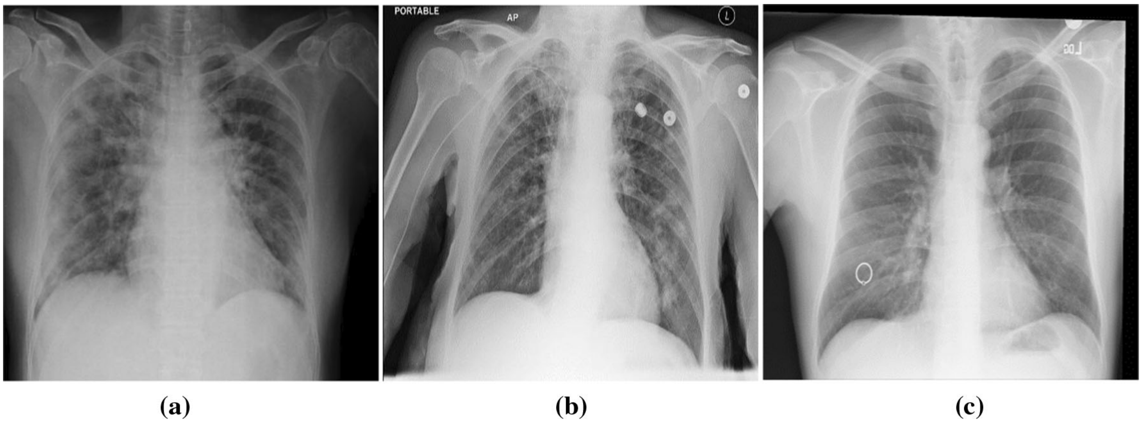


Fig. 1 Sample X-ray images in the data set: **a** COVID-19, **b** Pneumonia, **c** No-Findings

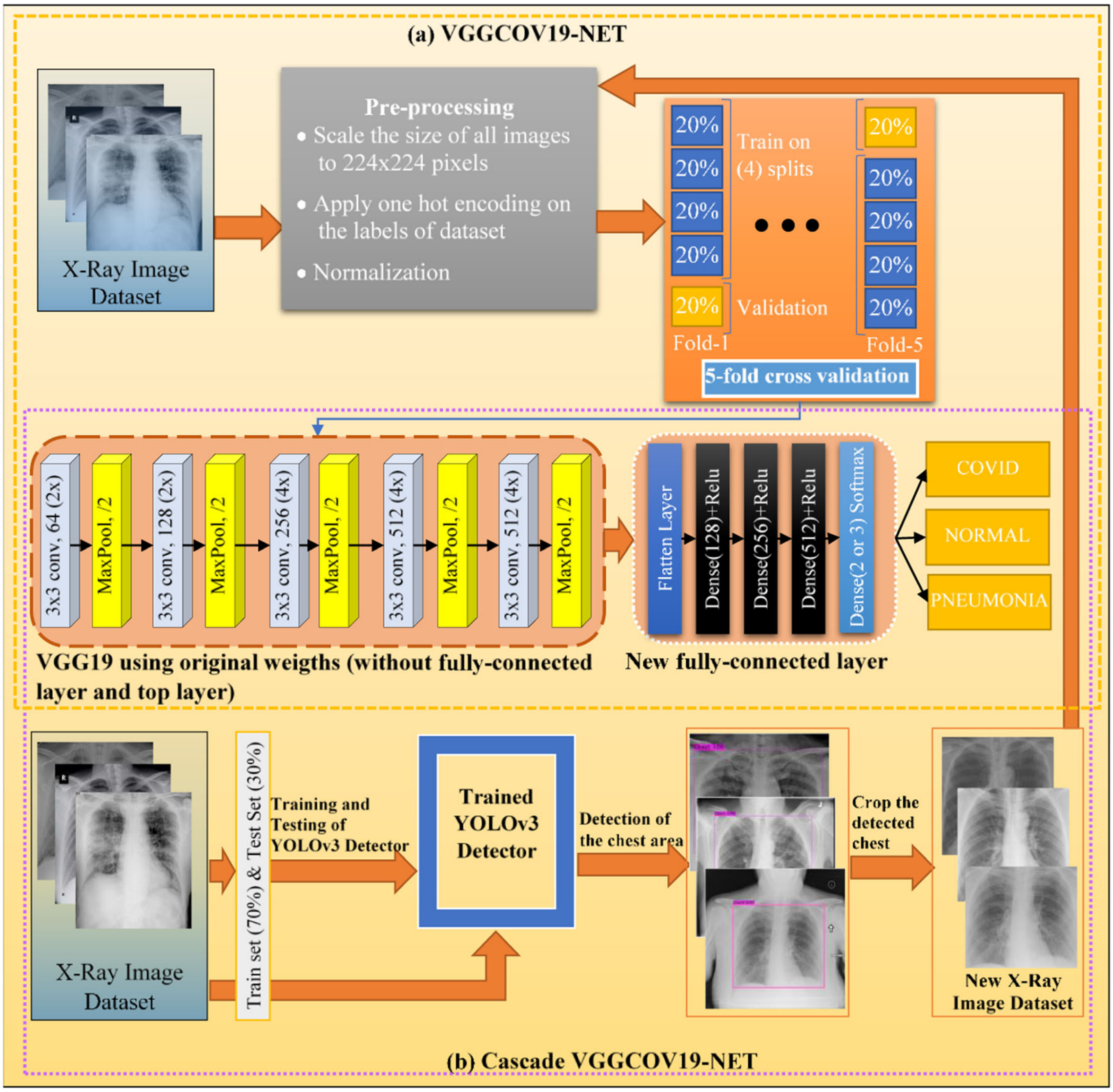


Fig. 2 Architectures of VGGCOV19-NET and Cascade VGGCOV19-NET models

number, and activation function in the FC layer were determined as a result of many trials.

The YOLOv3 detection algorithm is placed at the front of the VGGCOV19-NET and VGG19 models in the Cascade VGGCOV19-NET and Cascade VGG19 models. The YOLOv3 algorithm detects the chest zones in X-ray images and crops them so that a new X-ray data set focusing on the chest zone is attained. This data set is given to the VGGCOV19-NET and VGG19 models as inputs and the training and test processes of the Cascade models are conducted.

CNNs are designed using main multiple building blocks such as convolution, pooling, and FC layers [47]. The structure of the convolutional and pooling layers of the VGCOV19-NET model can be expressed as follows:

$$\begin{aligned}&CL1,\ CL2,\ PL1,\ PL2,\ CL3,\ CL4,\ PL3,\ PL4,\ CL5,\\&\quad CL6,\ CL7,\ CL8,\ PL5,\ PL6,\\&CL9,\ CL10,\ CL11,\ CL12,\ PL7,\ PL8,\ CL13,\ CL14,\\&\quad CL15,\ CL16,\ PL9,\ PL10.\end{aligned}$$

Here, CL refers to the convolutional layer and PL refers to the pooling layer using the maximum pooling method.

In the VGGCOV19-NET model, the convolution layer is very important as it performs the feature extraction process. This layer uses the convolution operation (represented by *) instead of the general matrix multiplication and is the

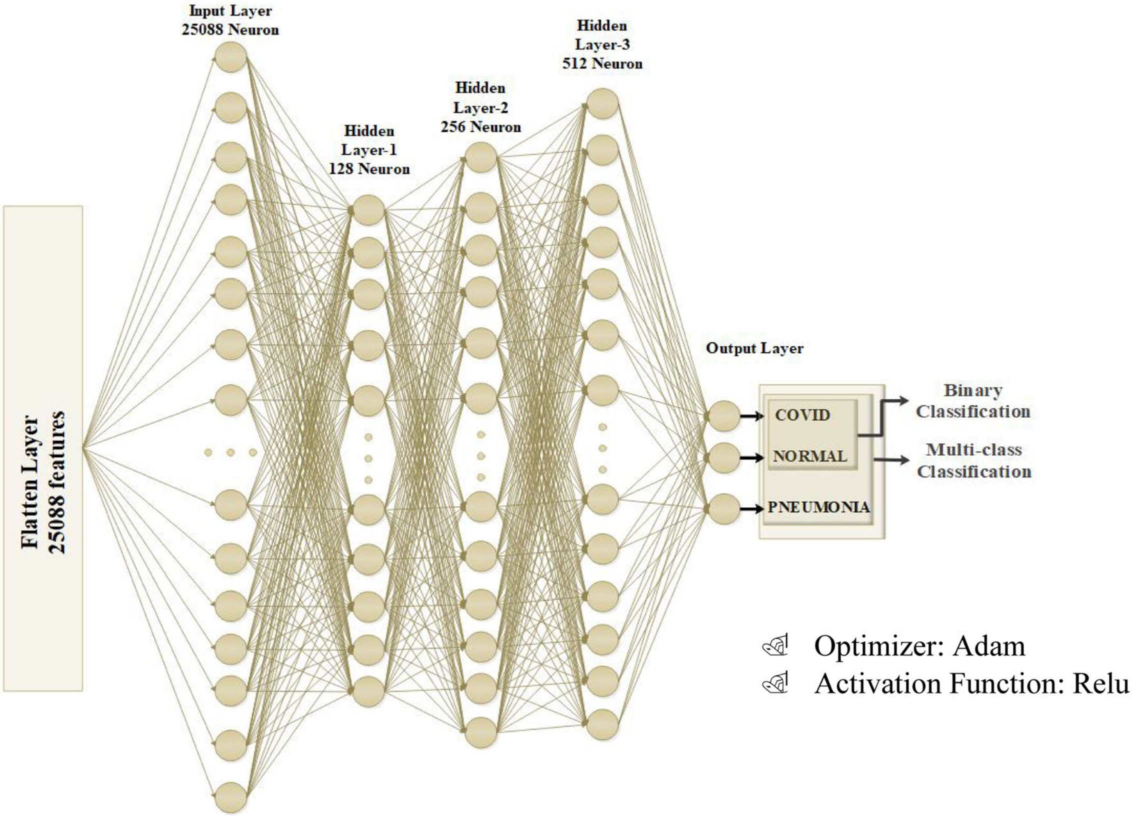


Fig. 3 Architecture of the FC layer of VGGCOV19-NET

core building block of CNN. The parameters of the convolution layer are made up of a set of learnable filters, also known as kernels. The main task of the convolutional layer is to identify the features in the local regions of the input image and to create a feature map [11].

This process is based on the method of wandering a selected filter on the input image. The dimension of the filter can be 3 × 3, 5 × 5 or 7 × 7 pixels. The input of the next layer ($m_2 \times m_3$) is formed with the filter applied to the image. Activation maps occur as a result of this convolution process. Activation maps have local distinctive features.

Each convolution layer has a filter (m_1). The output of Layer l consists of $m_1^{(l)}$ feature maps in $Y_i^{(l)}$, $m_2^{(l)} \times m_3^{(l)}$ dimension. The i. feature map shown with $Y_i^{(l)}$ is calculated according to Eq. 1. Here, $B_i^{(l)}$ represents the deviation matrix, and $K_{i,j}^{(l)}$ represents the filter dimension [48–50].

$$Y_i^{(l)} = f\left(B_i^{(l)} + \sum_{j=1}^{m_i^{l-1}} K_{i,j}^{(l)} \times Y_j^{(l-1)}\right) \tag{1}$$

Pooling layers gradually decrease the image dimension so the parameter number and calculation complexity of the model are decreased [51]. The pooling layer has two parameters, (l) the dimension of $F^{(l)}$ filter and $S^{(l)}$ step. This layer takes the data in the dimension of $m_1^{(l-1)} \times m_2^{(l-1)} \times m_3^{(l-1)}$ as input and again, provides $m_1^{(l)} \times m_2^{(l)} \times m_3^{(l)}$ output volume. The operation of the pooling layer is shown in Eqs. 2, 3 and 4 in short [49].

$$m_1^{(l)} = m_1^{(l-1)} \tag{2}$$

$$m_2^{(l)} = \left(m_2^{(l-1)} - F^{(l)}\right)/S^{(l)} + 1 \tag{3}$$

$$m_3^{(l)} = \left(m_3^{(l-1)} - F^{(l)}\right)/S^{(l)} + 1 \tag{4}$$

In the pooling process, a v vector is reduced to a single scaler f (v) with pooling process f. There are basically two pooling types: average pooling $f_{\mathrm{a}}(v) = \frac{1}{P}\sum_{i=1}^{p} v_i$, and max pooling $f_{\mathrm{m}}(v) = \max_i v_i$ [52]. Max pooling is used in the VGGCOV19-NET model.

FC layers turn the feature maps output of (1) final convolution or pooling layer into a unidimensional vector,

(2) bind to one or more dense layers (3) update the weights, (4) and give the estimation of the final classification [47]. FC layer mapping is a multi-layered sensor for $m_1^{(l-1)} \times m_2^{(l-1)} \times m_3^{(l-1)}$. When $(l-1)$ is considered as an FC layer, the process stages of the FC layer can be shown as in Eq. 5 [49].

$$Y_i^{(l)} = f\left(Z_i^{(l)}\right) \quad \text{with} \quad Z_i^{(l)} = \sum_{j=1}^{m_1(l-1)} w_{i,j}^{(l)} y_i^{(l-1)} \tag{5}$$

In addition, the activation function is used at the output of each layer of the VGGCOV19-NET model. This activation function is Softmax for the output layer of the model, and rectified linear unit (Relu) for other layers. Softmax and Relu activation functions are mathematically defined as in Eqs. 6 and 7, respectively. Softmax is used to normalize the probability vector x for any input [9, 53].

$$\text{soft max}(x_i) = \frac{\exp(x_i)}{\sum_{c=1}^{C} \exp(x_c)} \tag{6}$$

$$f(x) = \begin{matrix} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{matrix} \tag{7}$$

3.3 Training of the YOLOv3 detection algorithm and automatic cropping of the chest region

YOLOv3 is a deep CNN developed for object detection. The chest region on the chest X-ray image should be tagged before the training of the YOLOv3 model. The labelImg (https://github.com/tzutalin/labelImg) graphical image annotation tool was used for this. A text file was formed for each image using this tool, and the coordinates of the chest region taking place in the image were saved in this file. There are (X_0, Y_0, X_1, Y_1) points giving information about the exact location of height (cH), width (cW), and chest region of the images tagged in the text file. The normalization process is conducted to make the formed data set convenient for YOLOv3 architecture. With this process, the center point coordinates (X, Y), height (H), and width (W) data of the chest region tagged in the image are attained as in Eqs. 8 and 9 [54]. These data are shown in Fig. 4.

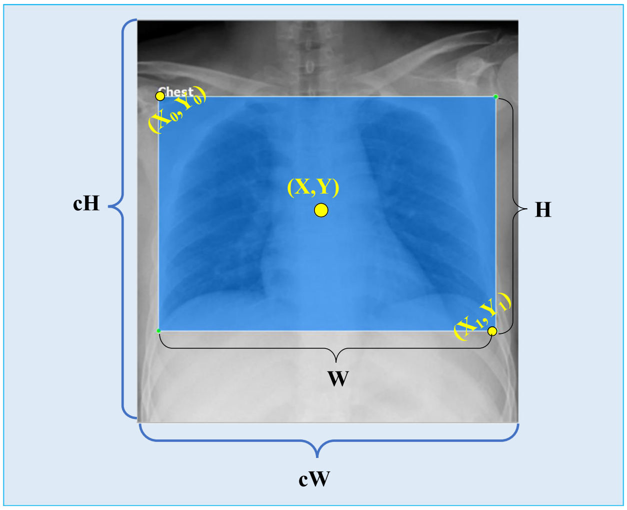


Fig. 4 Coordinates of the tagged chest X-ray image

$$X = \frac{X_1 + X_0}{2} \times \frac{1}{cW}, \quad Y = \frac{Y_1 + Y_0}{2} \times \frac{1}{cH} \tag{8}$$

$$W = (X_1 - X_0) \times \frac{1}{cW}, \quad H = (Y_1 - Y_0) \times \frac{1}{cH} \tag{9}$$

Training and test processes were conducted by randomly separating the data into two as 70% training and 30% test data after the completion of the preliminary processes. Model performance was evaluated according to Intersection over Union (IoU) and mean average Precision (mAP) metrics.

IoU is an important metric to reveal the model success in object detection. IoU calculates the similarity distance between the bounding box of the target and the estimated output [55]. The division of the region remaining between the estimated value and the necessary reference value to the whole region is attained as in Eq. 10 [54, 55].

$$\text{IoU} = \frac{\text{Overlap area between bounding box of target and predicted}}{\text{Combined area between bounding box of target and predicted}} \tag{10}$$

mAP ensures the evaluation of the values such as certainty, sensitivity, $F1$-score, and IoU from a single point. It is calculated as in Eq. 11 [54].

$$\text{mAP} = \int_0^1 P(R)\text{d}R \tag{11}$$

Processes were conducted with bounding boxes above a certain IoU threshold value while calculating the mAP value to determine the model performance. When the threshold value is taken as 0.50, mAP value for the test data was calculated as 99.80%, and the mean IoU value was calculated as 85.37. These values show that the Chest region was detected with high precision. The weights attained for these performance metrics were saved. Then, all the X-ray images in the data set were given to the trained YOLOv3 algorithm as input and the chest region was automatically detected. The detected chest region was cropped with OpenCv and a new data set containing the chest region was formed. The X-ray images whose Chest region were detected and cropped are shown in Fig. 5.

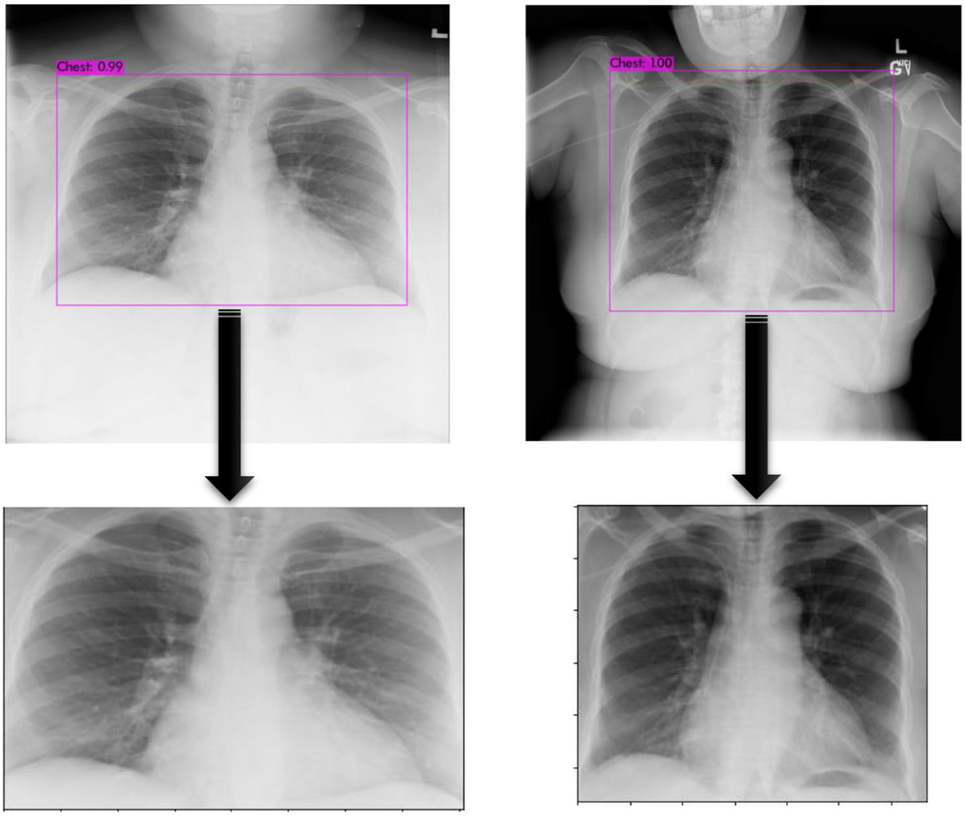


Fig. 5 Detection and cropping of the chest region in X-ray images

3.4 Training of VGGCOV19-NET and cascade VGGCOV19-NET models

VGGCOV19-NET and Cascade VGGCOV19-NET models were formed using Tensorflow and the Keras library in the Phyton programming language. Some preliminary processes were conducted on the data sets before the training. Firstly, the images were adjusted to the dimensions 224 × 224 × 3. Then, each input datum was divided by 255, and the normalization process was completed. One hot encoding process was applied on the output class labels. Different optimizer algorithms (Adam, Adadelta, Sgd, RMSprop, Adamax, Nadam), loss functions (binary cross-entropy, categorical cross-entropy), and learning rates were tried to train the models. The best classification performance was attained for Adam optimizer, categorical cross-entropy loss function and 0.001 learning rate. Pre-trained weights of the VGG19 model were used in convolution layers and the trainable properties of these layers were passivized so decreasing calculation time and cost. The fivefold cross-validation method was used in the training and testing of the models. Models were trained by 100 epochs for each fold, and a high level of classification performance was achieved. The formation and training of CNN models without using GPU is very hard. For this reason, the training process of models was conducted using T4 and P100 GPUs on Google Colab (a product of Google Research). In addition, the original VGG19 model was also trained under similar conditions, and classification performance was attained. Thus, the difference in the classification performance of the VGGCOV19-NET model was better determined. The pseudocode carrying out the processes of forming the model explained in Sect. 3.2 and training the model explained in this section is given below. This code clearly reveals the formation and training of VGGCOV19-NET and Cascade VGGCOV19-NET models.

Pseudo-code of VGGCOV19-NET and Cascade VGGCOV19-NET models

```
X_YOLO (X-ray Images with Chest Region Detected and Croped) ← Trained YOLOv3 ← X-ray Images
Which-Model ← 0 or 1 (0:VGGCOV19-NET,  1:Cascade VGGCOV19-NET)
If (Which-Model==0) Input: X← X-ray images in 224x224x3 dimension
Else Input: X←X_YOLO X-ray images in 224x224x3 dimension
Output: Y← categories (Normal, Covid-19, Pneumonia)
X←X/255 (normalize input data)
Y← OneHoteEncoding(Y) (Normal=0, Covid-19=1, Pneumonia=2)
vgg ←VGG19(), build VGG-19 architecture without fully connected layer and top layer
vgg_layer_list ←vgg.layers
//---------Build VGGCOV19-NET model------------
model ←Sequential()
For layer in vgg_layer_list do
   model← layer
End for
For layer in model.layers do
   layer.trainable=False
End for
numberOfClass ← 2 or 3
model ← Flatten()
model ← Dense(128)
model ← Dense(256)
model ← Dense(512)
//-------------------------------------------------------------------
//--------- Perform training and testing of VGGCOV19-NET model------------------
kf = KFold(n_splits=5)
For train_index, test_index in kf.split(X) do
    x_train ← X[train_index],  y_train ← Y[train_index]
    x_test ← X[test_index],    y_test ← Y[test_index]
    model.compile(loss="categorical_crossentropy", optimizer="Adam", metrics=["accuracy"])
    history ← model.fit(x_train, y_train, epochs=100)
    Calculate performance metrics (recall, accuracy, precision, auc, f1-score, confusion matrix) for each Fold
End for
Output: VGGCOV19-NET CNN classifier and classification results
```

Two different scenarios were tested to diagnose COVID-19 using X-ray images. While the VGGCOV19-NET, VGG19, and Cascade models had initially been trained to classify three categories (COVID-19 vs. no-findings vs. pneumonia), they were trained to classify two categories (COVID-19 vs. no-findings) in the second scenario. In other words, multi-class classification was conducted in scenario-1 and binary class classification in scenario-2. Fivefold cross-validation was used in the training and test processes of the models. Training accuracy and training loss values of multi-class and binary classifier models are shown in Figs. 6 and 7 for the fold-2 step. It was observed that the Cascade VGGCOV19-NET model showed a faster training process than the other models for multi-class. When loss values were examined for the multi-class classification, the Cascade VGGCOV19-NET model decreased the loss value very rapidly and approached zero value. However, the loss value decreasing performances of VGGCOV19-NET and VGG19 models were low, inconsistent and unable to approach zero value. It was also noted that the VGG19 model had many fluctuations in both the loss and accuracy graphs. This is an indication that the VGG19 model is unstable in the learning process and does not perform good learning. In contrast to the VGG19 model, the Cascade VGG19 model conducted stable learning in the multi-class classification. All models conducted stable learning in the binary classification, and the difference between their learning speeds was a little lower and similar to the multi-class classification.

The training, testing times, and total trainable parameters of the models are shown in Table 1. The lowest time belongs to the binary Cascade VGGCOV19-NET model. The highest time belongs to the multi-class VGG19 model. In addition, the training and testing times of the Cascade models are significantly lower than the other models. While this rate was approximately 70% in the binary classification, it was approximately 60% in the multi-class classification. The reason for this decrease in training and testing

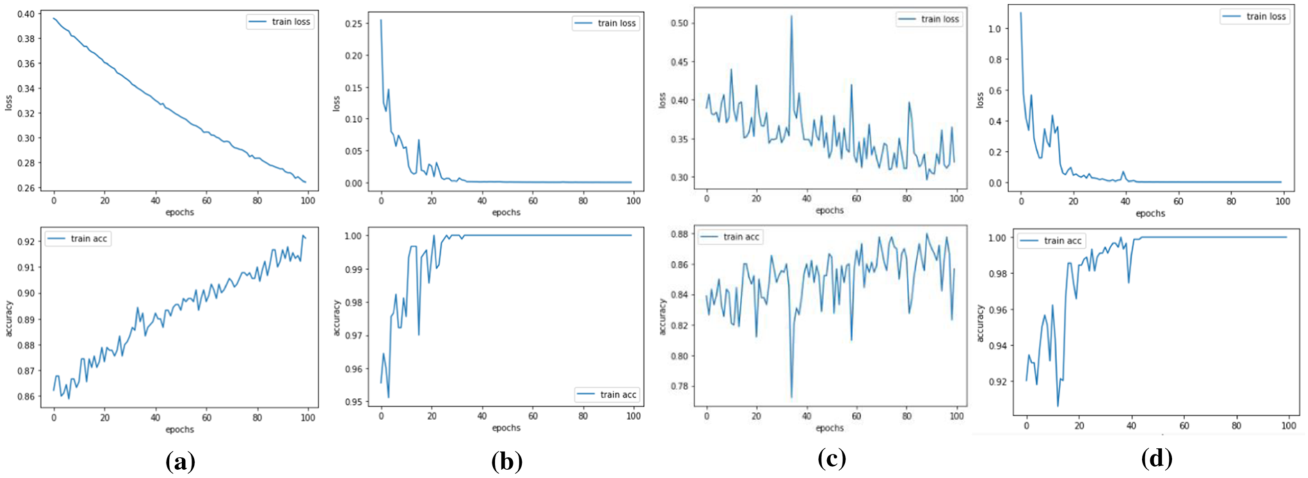


Fig. 6 Training loss and accuracy curves attained at fold-2 for multi-class classification: **a** training loss and accuracy curves of the VGGCOV19-NET model, **b** training loss and accuracy curves of the Cascade VGGCOV19-NET model, **c** training loss and accuracy curves of the VGG19 model, **d** training loss and accuracy curves of the Cascade VGG19 model

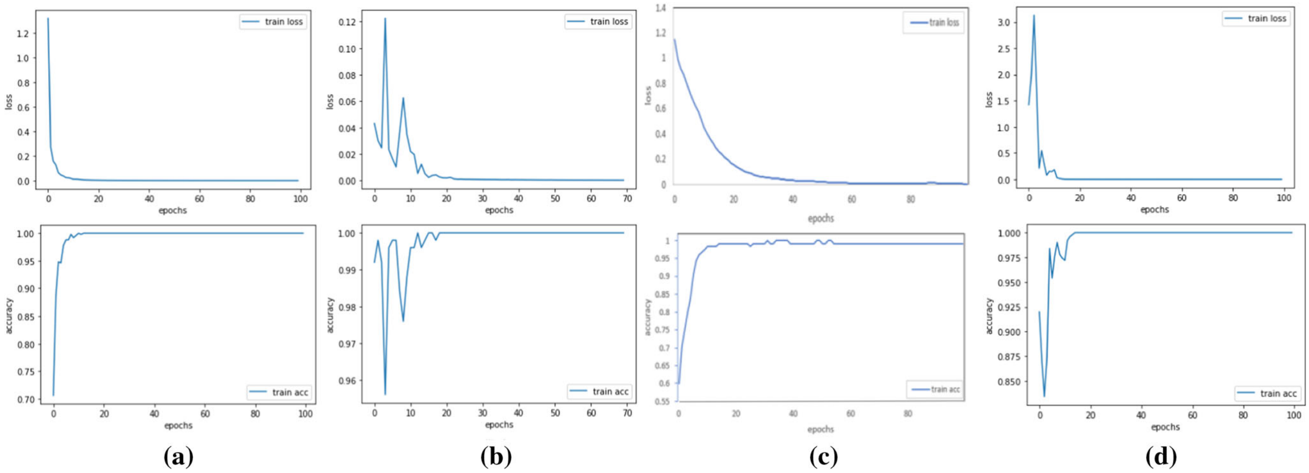


Fig. 7 Training loss and accuracy curves attained at fold-2 for binary classification: **a** training loss and accuracy curves of the VGGCOV19-NET model, **b** training loss and accuracy curves of the Cascade VGGCOV19-NET model, **c** training loss and accuracy curves of the VGG19 model, **d** training loss and accuracy curves of the Cascade VGG19 model

Table 1 Training and testing times of the models and numbers of trainable parameters

Classification type	Model	Training and testing time (minutes)	Number of trainable parameters (*M*)
Binary class	VGGCOV19-NET	14.27	3.37
	Cascade VGGCOV19-NET	4.13	
	VGG19	18.29	119.55
	Cascade VGG19	6.60	
Multi-class	VGGCOV19-NET	25.36	3.37
	Cascade VGGCOV19-NET	10.38	
	VGG19	32.57	119.55
	Cascade VGG19	11.75	

time is that Cascade models focus only on the chest region and decrease the numbers of pixels to be processed. Also, the training and testing times of the VGGCOV19-NET models were lower than those of the VGG19 models. Similarly, the trainable parameters of the VGGCOV19-NET model are lower because the neuron number in the FC layer is low.

3.5 Performance metrics

Models were evaluated according to various performance metrics as follows: precision (*P*), recall (*R*), specificity (*S*), *F*1-score (*F*1), area under the receiver operating characteristics (ROC) curve (AUC), and accuracy (acc).

- *Precision* This parameter measures the proportion of anticipated positives that are true positives. Therefore, it is dependent on true positive (TP) and false positive (FP) values [56].

$$P = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP}) \tag{12}$$

- *Recall* Recall is the ratio of the true positives classified correctly by the model [57]. TP and FN values are used to calculate Recall [56].

$$R = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}) \tag{13}$$

- *Specificity* Specificity is the ratio of the true negatives (those not from pathology) classified correctly by the model [57]. TN and FP values are used to calculate specificity [36].

$$S = \mathrm{TN}/(\mathrm{TN} + \mathrm{FP}) \tag{14}$$

- *F1-Score* *F*1-score is an overall measure of the model's accuracy that combines precision and recall. It is the double of the ratio of the multiplication of *F*1-score precision and recall metrics to their totals [16].

$$F1 = 2 \times (P \times R)/(P + R) \tag{15}$$

- *AUC* ROC curves are widely used to reveal the results for binary classification problems in machine learning [58]. The ROC curve verifies the classification performance by showing the false positive rate (FPR) against the true positive rate (TPR). AUC is an important parameter for assessing the discriminative ability of prediction models [16, 59].

4 Results

In this part, the results obtained from VGGCOV19-NET, Cascade VGGCOV19-NET, VGG19 and Cascade VGG19 models are presented in detail. As specified above, the VGGCOV19-NET is a model developed by the reformation of the full connected layer of VGG19 CNN architecture. Cascade models are the ones formed by feeding the input of VGGCOV19-NET and VGG19 models with the output of the YOLOv3 algorithm.

R, *P*, *F*1, and acc metrics whose mathematical formulations were explained in the methods part were used to evaluate the classification performance of the models. The acc metric determines at what ratio a model anticipates the values. The *P* metric determines the repeatability of the measurement or how many anticipations are true. *R* represents the number of correct results found. The *F*1 score uses precision and recall metrics to calculate an average result [38]. Both multi-class and binary classification performances of the models were evaluated according to these metrics for each fold. Also, an overlapped confusion matrix (CM) was formed for the purpose of conducting a general evaluation of the models and the performance metrics generally representing the model were calculated using this matrix. To create the overlapped CM, the CMs obtained from all the folds are summed [9].

For multi-class classification, the classification performances of VGGCOV19-NET, Cascade VGGCOV19-NET, VGG19 and Cascade VGG19 models are given in Table 2, and overlapped CM is given in Fig. 8. As seen in this table, the weighted average values of *R*, *P*, and *F*1 metrics are given for each fold. However, the metric values presented under the line "overlapped" have been calculated separately for each category from the overlapped CM, and these values are not weighted averages. Consequently, the use of these values will provide more explanatory data in the revelation of the general performances of the models for each category.

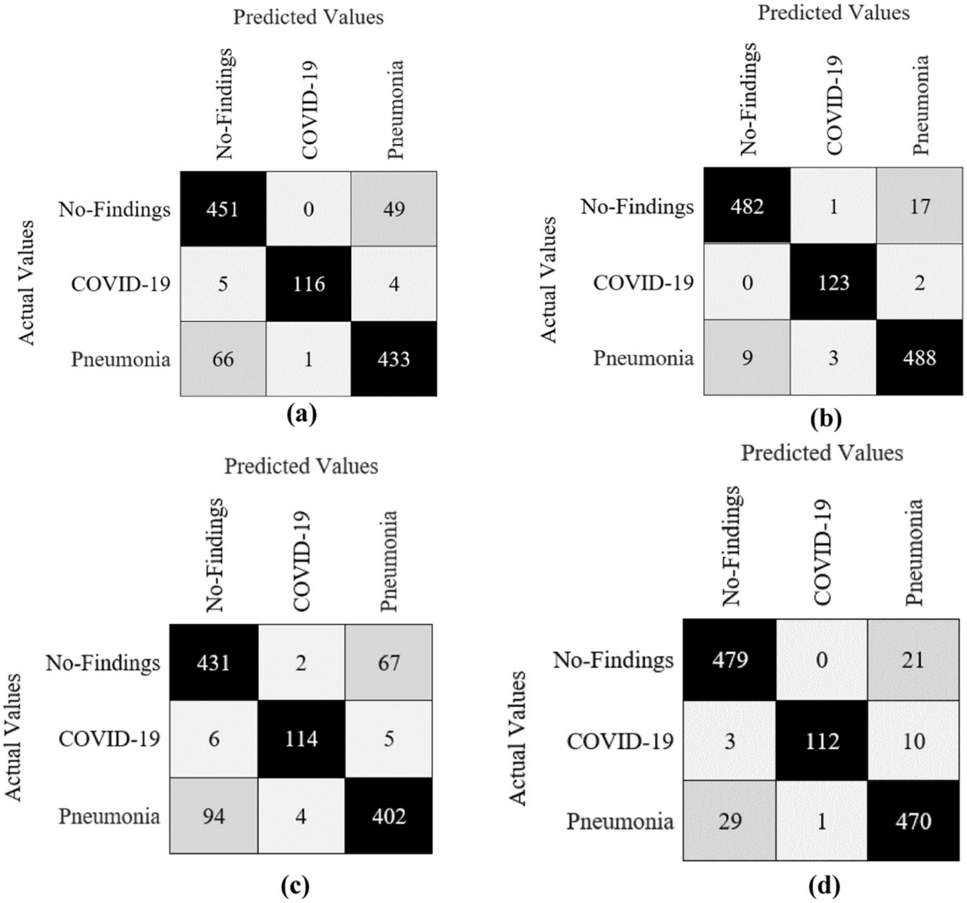
When an evaluation is conducted according to the acc metric, the classification performance of the Cascade VGGCOV19-NET model is seen to be much higher than that of the VGGCOV19-NET, VGG19 and Cascade VGG19 models. The highest mean accuracy value (acc = 97.16%) belongs to the Cascade VGGCOV19-NET model. The Cascade version of VGGCOV19-NET model increased the classification performance. A similar situation also exists between the VGG19 model (acc = 84.17%) and the Cascade VGG19 model (acc = 94.31%). However, the increase in the classification performance is far higher here. Also, according to the *R* metric, the Cascade VGGCOV19-NET model correctly classified the COVID-19, pneumonia, and no-findings patients to an extremely high extent. The Cascade VGGCOV19-NET model correctly classified COVID-19 patients by 98.40% (*R* = 0.9840), pneumonia patients by 97.60% (*R* = 0.9760), and no-findings patients by 96.40% (*R* = 0.9640). These figures are eye-catching for model conducting multi-class classifications. VGGCOV19-NET model correctly classified the COVID-19 patients by 92.80% (*R* = 0.9280),

Table 2 R, P, $F1$-score, and acc values for COVID-19, no-findings, and pneumonia classes of the VGGCOV19-NET, Cascade VGGCOV19-NET, VGG19, and Cascade VGG19 models

Models	Fold	Performance results (%)			
		Recall	Precision	$F1$	Accuracy
VGGCOV19-NET	Fold 1	78.20*	78.80*	78.30*	78.22
	Fold 2	91.10*	91.10*	91.10*	91.11
	Fold 3	84.40*	84.80*	84.50*	84.44
	Fold 4	95.10*	95.20*	95.10*	95.11
	Fold 5	95.60*	95.70*	95.60*	95.56
	Overlapped				
	COVID-19	92.80	99.15	95.87	
	No-findings	90.20	86.40	88.26	
	Pneumonia	86.60	89.09	87.83	
	Average	89.87	91.55	90.65	88.89
Cascade VGGCOV19-NET	Fold 1	85.80*	86.30*	85.90*	85.80
	Fold 2	100*	100*	100*	100
	Fold 3	100*	100*	100*	100
	Fold 4	100*	100*	100*	100
	Fold 5	100*	100*	100*	100
	Overlapped				
	COVID-19	98.40	96.85	97.62	
	No-findings	96.40	98.17	97.28	
	Pneumonia	97.60	96.25	96.92	
	Average	97.47	97.09	97.27	97.16
VGG19	Fold 1	80.40*	80.40*	80.20*	80.44
	Fold 2	80*	81.50*	79.80*	80
	Fold 3	85.30*	87*	85.20*	85.33
	Fold 4	88*	88.70*	87.90*	88
	Fold 5	87.10	87.40*	87*	87.11
	Overlapped				
	COVID-19	91.20	95	93.06	
	No-findings	86.20	81.16	83.60	
	Pneumonia	80.40	84.81	82.54	
	Average	85.93	86.99	86.40	84.17
Cascade VGG19	Fold 1	83.10*	83.60*	83.10*	83.11
	Fold 2	99.10*	99.10*	99.10*	99.11
	Fold 3	93.30*	93.50*	93.30*	93.33
	Fold 4	97.30*	97.40*	97.30*	97.33
	Fold 5	98.70*	98.70*	98.70*	98.67
	Overlapped				
	COVID-19	89.60	0.9912	0.9412	
	No-findings	95.80	0.9374	0.9476	
	Pneumonia	94.00	0.9381	0.9391	
	Average	93.13	0.9556	0.9426	94.31

*Weighted average

pneumonia patients by 86.60% (R = 0.8660), and no-findings patients by 90.20% (R = 0.9020). In addition, the Cascade VGGCOV19-NET model correctly detected COVID-19 patients at a higher rate than pneumonia and no-findings patients. The VGG19 model correctly classified COVID-19 patients by 91.20% (R = 0.9120), pneumonia patients by 80.40% (R = 0.8040), and no-findings patients by 86.20% (R = 0.8620). These results are, respectively, 89.60%, 94%, and 95.80% in the Cascade VGG19 model. The increase in the correct classification



Actual Values \ Predicted Values	No-Findings	COVID-19	Pneumonia
No-Findings	451	0	49
COVID-19	5	116	4
Pneumonia	66	1	433

(a)

Actual Values \ Predicted Values	No-Findings	COVID-19	Pneumonia
No-Findings	482	1	17
COVID-19	0	123	2
Pneumonia	9	3	488

(b)

Actual Values \ Predicted Values	No-Findings	COVID-19	Pneumonia
No-Findings	431	2	67
COVID-19	6	114	5
Pneumonia	94	4	402

(c)

Actual Values \ Predicted Values	No-Findings	COVID-19	Pneumonia
No-Findings	479	0	21
COVID-19	3	112	10
Pneumonia	29	1	470

(d)

Fig. 8 Overlapped confusion matrices of models for multi-class classification **a** overlapped confusion matrix of the VGGCOV19-NET model, **b** overlapped confusion matrix of the Cascade VGGCOV19-NET model, **c** overlapped confusion matrix of the VGG19 model, **d** overlapped confusion matrix of the Cascade VGG19 model

ratio in the Cascade version of VGG19 model is significant for the three categories.

The second scenario used to diagnose COVID-19 is the binary classification. Sensitivity, specificity, precision, $F1$-score, auc, accuracy, and confusion matrix results of the models for binary classification are given in Table 3. The average accuracy values of the Cascade VGGCOV19-NET, VGGCOV19-NET, Cascade VGG19, and VGG19 models are 99.84%, 98.72%, 99.04%, and 98.56%, respectively. The classification precision of the Cascade VGGCOV19-NET model is slightly higher. The Cascade versions of the VGGCOV19-NET and VGG19 models fed with YOLOv3 algorithm also increased the classification performance for the binary classification. Revelation of how correctly the models classify the true positives (TP) and true negatives (TN) according to R and S metrics is important in binary classification. While the Cascade VGGCOV19-NET model correctly classified the COVID-19 patients by 99.20%, it correctly classified no-findings patients by 100%. It is striking in binary classification for the model to correctly classify both true positives and false positives at such a high ratio. In addition, while the VGGCOV19-NET model correctly classified the COVID-19 patients by 95.40%, it correctly classified no-findings patients by 99.60%. The provision of the VGGCOV19-NET model with Cascade structure caused an increase in the correct classification ratio (R = 0.9920) of COVID-19 patients. A similar situation exists in the Cascade VGG19 model. While the Cascade VGG19 model correctly classifies COVID-19 patients by 96.80%, the VGG19 model correctly classifies them by 92.80%. It could be said that feeding the VGG19 model with the YOLOv3 algorithm provided a greater increase in the correct classification ratio of COVID-19 patients.

Another metric used in the verification of the classification performance of the models is the ROC curve. ROC curves in each fold step and AUC values are shown in Fig. 9 for binary classification. The closeness of the AUC value to one shows that the classification performance of the model is high. The fact that the classification

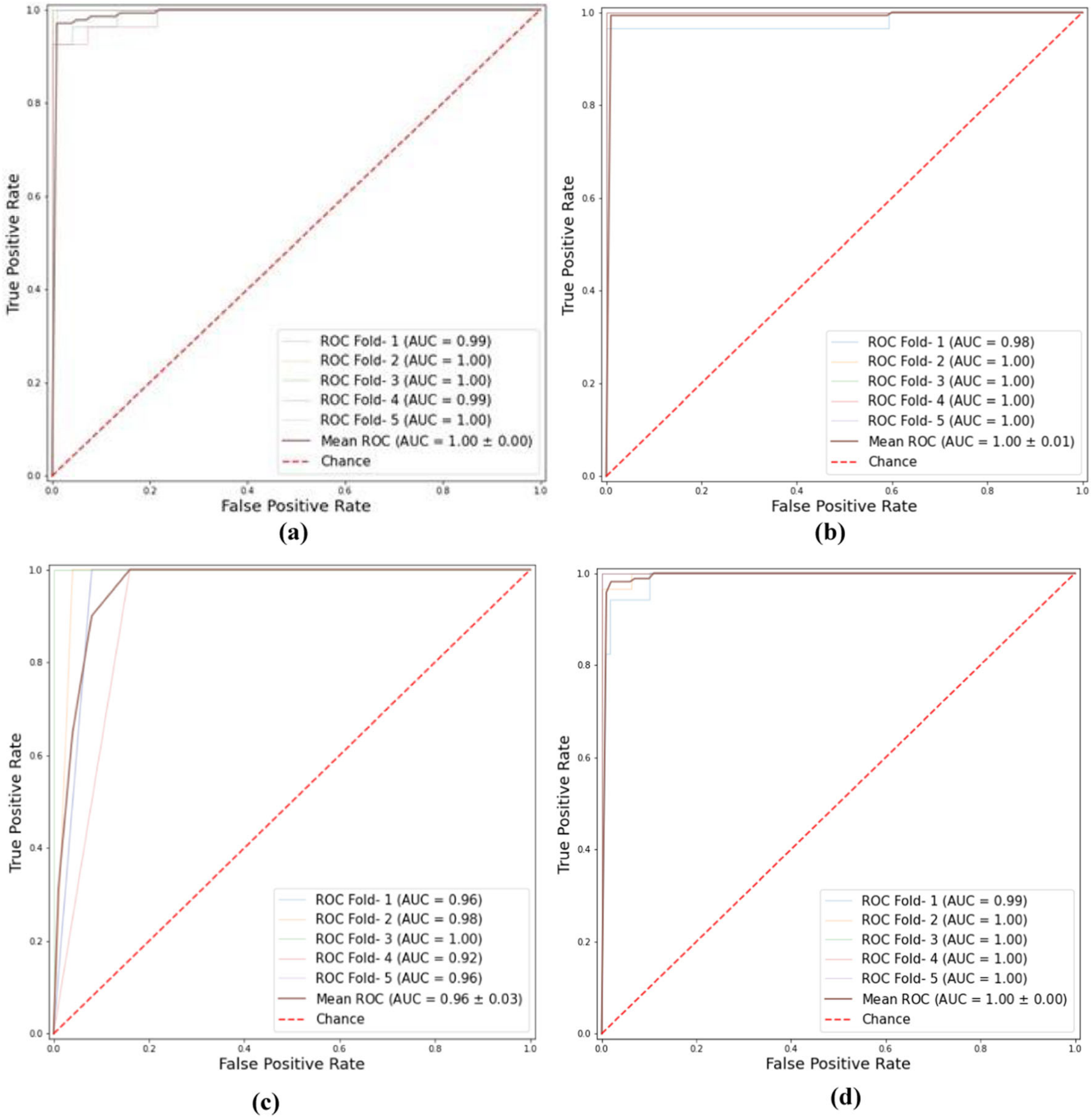
Table 3 Specificity, sensitivity, auc, $F1$-score, precision, and accuracy values for no-findings and COVID-19 classes of the VGGCOV19-NET, Cascade VGGCOV19-NET, VGG19, and Cascade VGG19 models

Models	Fold	Confusion matrix and performance results (%)									
		TP	TN	FP	FN	Recall	Sensitivity	Precision	$F1$	Accuracy	AUC
VGGCOV19-NET	Fold 1	24	98	0	3	89	100	98*	98*	97.60	99.12
	Fold 2	21	103	1	0	100	99	99*	99*	99.20	100
	Fold 3	28	97	0	0	100	100	100*	100*	100	100
	Fold 4	25	97	1	2	93	99	98*	98*	97.60	98.94
	Fold 5	21	103	0	1	95	100	99*	99*	99.20	99.95
	Overlapped										
	COVID-19	119	498	2	6	95.40	99.60	98.35	96.85		
	No-findings	498	119	6	2	99.60	95.40	98.81	99.20		
	Average					97.50	97.50	98.58	98.03	98.72	99.60
Cascade VGGCOV19-NET	Fold 1	28	96	0	1	0.97	1	0.99*	0.99*	99.20	98
	Fold 2	21	104	0	0	100	100	100*	100*	100	100
	Fold 3	29	96	0	0	100	100	100*	100*	100	100
	Fold 4	26	99	0	0	100	100	100*	100*	100	100
	Fold 5	20	105	0	0	100	100	100*	100*	100	100
	Overlapped										
	COVID-19	124	500	0	1	99.20	100	100	99.60		
	No-findings	500	124	1	0	100	99.20	99.80	99.90		
	Average					99.60	99.60	99.90	99.75	99.84	99.60
VGG19	Fold 1	23	100	0	2	92	100	98*	98*	98.40	96
	Fold 2	24	100	0	1	96	100	99*	99*	99.20	98
	Fold 3	25	100	0	0	100	100	100*	100*	100	100
	Fold 4	21	100	0	4	84	100	97*	97*	96.80	92
	Fold 5	23	100	0	2	92	100	98*	98*	98.40	96
	Overlapped										
	COVID-19	116	500	0	9	92.80	100	100	96.26		
	No-findings	500	116	9	0	100	92.8	98.23	99.10		
	Average					96.40	96.40	99.11	97.68	98.56	96.40
Cascade VGG19	Fold 1	16	106	1	2	94	98	98*	98*	97.60	98
	Fold 2	27	96	2	0	93	100	98*	98*	98.40	99.78
	Fold 3	25	99	1	0	96	100	99*	99*	99.20	99.80
	Fold 4	30	95	0	0	100	100	100*	100*	100	100
	Fold 5	23	102	0	0	100	100	100*	100*	100	100
	Overlapped										
	COVID-19	121	498	4	2	96.80	99.60	98.37	97.58		
	No-findings	498	121	2	4	99.60	96.80	99.20	99.40		
	Average					98.20	98.20	98.79	98.49	99.04	99.52

*Weighted average

performances of VGGCOV19-NET, Cascade VGGCOV19-NET, and Cascade VGG19 models are very high was also verified by the average AUC values (99.60%, 99.60%, and 99.52%). The average AUC value (96.40%) of the VGG19 model is lower compared to these models.

Two hundred and nineteen different COVID-19 X-ray images were [60] used to verify the performance of the Cascade VGGCOV19-NET model which yielded the best performance in the study. No different data sets were used for no-findings and pneumonia. The Cascade VGGCOV19-NET model displayed better performance for this new dataset in both binary and three-class classification. The accuracy value was 99.18% for three-class classification and 99.58% for binary classification. These values are



(a)

(b)

(c)

(d)

Fig. 9 ROC curves of the models for binary classification **a** ROC curve of the VGGCOV19-NET model, **b** ROC curve of the Cascade VGGCOV19-NET model, **c** ROC curve of the VGG19 model, **d** ROC curve of the Cascade VGG19 model

96.56% and 98.61%, respectively, for VGGCOV19-NET. These results prove that the YOLO-aided Cascade architecture increases the classification performance.

5 Discussion

The accuracy values of the models are shown comparatively in Fig. 10 for binary and multi-class classifications. As can be seen in this figure, the classification performance of Cascade VGGCOV19-NET is higher than that of the other models for both multi-class and binary classifications. Moreover, the Cascade VGGCOV19-NET model generally predicts COVID-19 at high ratios.

For the Cascade VGGCOV19-NET model to have high accuracy values in binary and multi-class classification is an indicator that the model has been structured well. Furthermore, the Cascade VGGCOV19-NET model classifies COVID-19 correctly by 98.40% (R = 0.9840) in the multi-class classification in which there are three categories. This ratio is lower in the VGGCOV19-NET model (R = 0.9280) than the Cascade VGGCOV19-NET model. The fact that the recall parameter is high in the Cascade VGGCOV19-NET model proves that they can correctly distinguish COVID-19 patients from pneumonia patients and no-findings patients at a high ratio. In addition, the fact that making some changes to the original VGG CNN architecture increases the model performance is also shown in

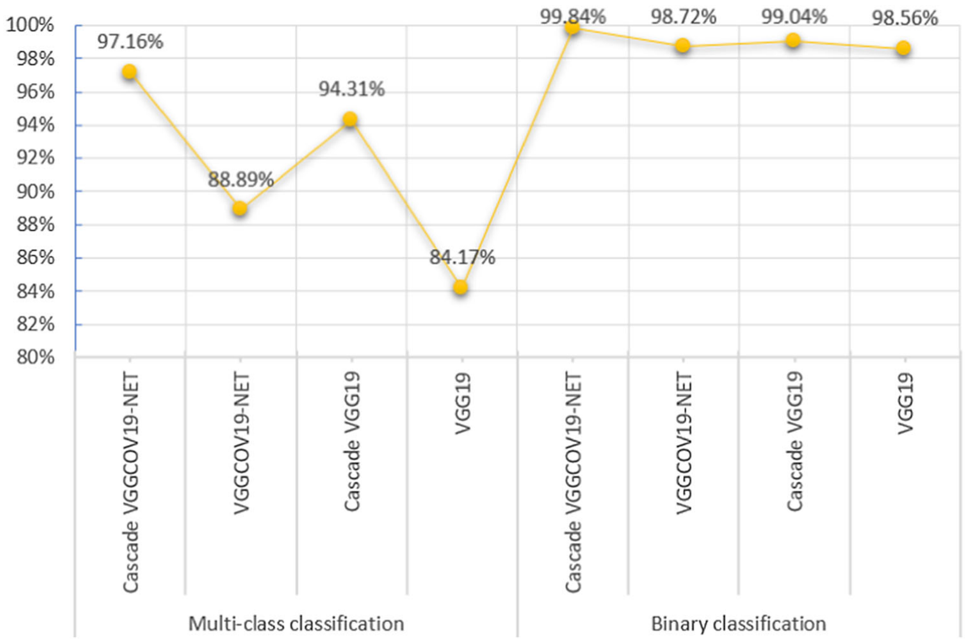


Fig. 10 Comparison of the accuracy values of the models

this study. Similarly, it also revealed that instead of using pre-trained CNN models alone, the formation of a Cascade structure via feeding with algorithms such as the YOLO detection algorithm increases the classification performance. A comparison of the model performance attained in this study to previous studies is also important. These results are given in Table 4.

According to this table, the classification performance of this study is higher than other studies in the literature both for binary and three-class classifications. DarkCovidNet developed by Ozturk et al. [9] correctly classified the binary class with an accuracy of 98.08% and three-class by 87.02%. DarkCovidNet is based on DarkNet architecture where fewer layers and filters are used. The data set used in that study is the same as the one used in our study. The classification performance of Cascade VGGCOV19-NET and Cascade VGG19 developed in our study is very much higher than that of DarkCovidNet. This performance difference is approximately 9.5% for the three-class classification and 1.28% in the binary classification. There are two studies [11, 37] showing the closest performance to Cascade VGGCOV19-NET and Cascade VGG19 among other studies using different data sets for binary classification. The accuracy values of both studies are 99%. While the first of these studies [11] performed the classification process with a new Xception CNN architecture-based CNN model on 284 COVID-19 positive and 310 normal X-ray images, the second [37] performed the classification process using pre-trained CNN InceptionV3 architecture and the transfer learning approach on 160 COVID-19 (+) and 160 normal X-ray images. There are also studied in the literature classifying three classes. Khan et al. [11] classified three classes with a new Xception CNN architecture-based CNN model with an accuracy ratio of 95%, Ahammed et al. [34] classified three classes with the deep CNN model they created with accuracy of 94%, Apostolopoulos and Mpesiana [35] classified three classes with the pre-trained VGG19 architecture with accuracy of 93.48%, Aslan et al. [30] classified three classes with the hybrid CNN architecture with accuracy of 98.70%, Hira et al. [31] classified three classes with the specially ResNeXt-50 with accuracy of 97.55%, and Wang et al. [29] classified three classes with the deep CNN model they called COVID-Net with an accuracy of 93.3%. The Cascade VGGCOV19-Net model suggested in our study classify three classes with accuracy higher than these studies, except for two studies [30, 31].

Furthermore, there are two studies using the YOLO architecture in the literature. Nigam et al. [62] tried to detect COVID-19 from X-ray images. They obtained the highest classification performance for three-class classification in the EfficientNet model with 93.48%. First, X-ray images are given to the YOLO algorithm in this study to determine the lung zone as in our study. Then, these images are classified with pre-trained CNN models. The data set used in the study is a large data set. However, the classification accuracy is lower than our study. Furthermore, it could not be demonstrated whether the YOLO-based model affected the performance as the study was not conducted on raw images. Al-antari et al. [61] tried to detect nine lung diseases including COVID-19 and pneumonia. Unlike our study, they only used the YOLO detection algorithm through X-ray images whose ground truth was labeled. The average detection accuracy for all classes was 90.67%.

Table 4 Comparison of the recommended VGGCOV19-NET COVID-19 diagnosis method with other CNN methods developed using radiology images

Study	Type of images	Number of cases	Method used	Segmentation or location attention	Accuracy (%)	Remarks
Ozturk et al. [9]	Chest X-ray	500 No-findings 125 COVID-19 (+)	DarkCovidNet	–	98.08	Although the same data set as our study was used, the classification performance is significantly lower for three-class classification
		500 No-findings 500 Pneumonia 125 COVID-19 (+)			87.02	
Hemdan et al. [16]	Chest X-ray	25 COVID-19 (+) 25 Normal	VGG19 and DenseNet201	–	90	The number of COVID-19 X-ray images in the data set is very low. Only binary classification was performed and the classification performance is considerably low
Wang et al. [29]	Chest X-ray	358 COVID-19 (+) 5538 Pneumonia 8066 Normal	COVID-Net	–	93.3	They reduced the computational complexity with the CNN architecture where they used 1 × 1 convolution blocks. The accuracy value is lower compared to our study and the sensitivity value needs improving
Khan et al. [11]	Chest X-ray	284 COVID-19 (+) 657 Pneumonia 310 Normal	CoroNet based on pre-trained Xception CNN	–	95	The classification performance is relatively lower for three-class classification
		284 COVID-19 (+) 310 Normal			99	
Medhi et al. [32]	Chest X-ray	More than 150 COVID-19 (+)	CNN	–	93	The data set is not clear. They used the CNN model they had developed themselves. They did not apply the pre-trained models. Furthermore, the binary classification performance is low
Harit et al. [33]	Chest X-ray	46 COVID-19 (+) 41 normal	ResNet50	–	94	The data set is very small. Moreover, the classification for binary class is considerably low. Only the ResNet50 pre-trained model was applied
Ahammed et al. [34]	Chest X-ray	219 COVID-19 (+) 1345 Viral pneumonia 1341 Normal	CNN, ResNet50, VGG16, InceptionV3and traditional machine learning methods	–	94	They attained the highest performance with the CNN developed by themselves. Low performance was attained with traditional methods. Classification accuracy is relatively low

Table 4 (continued)

Study	Type of images	Number of cases	Method used	Segmentation or location attention	Accuracy (%)	Remarks
Apostolopoulos and Mpesiana [35]	Chest X-ray	224 COVID-19 (+) 700 Pneumonia 504 Healthy	VGG19	–	93.48	The VGG19 pre-trained CNN architecture was used in raw form without being modified. The classification performance is significantly lower compared to our study especially for three-class classification
		224 COVID-19 (+) 504 Healthy			98.75	
Narin et al. [36]	Chest X-ray	50 COVID-19 (+) 50 COVID-19 (−)	ResNet50 and deep CNN	–	98	Only binary classification was performed in the study and the number of X-ray images in the data set is very low
Ouchicha et al. [17]	Chest X-ray	219 COVID-19 1341 normal 1345 viral pneumonia	CNN based on ResNet	–	96.69	A new CNN model based on the ResNet model was proposed. Although the classification performance for three-class classification is higher than other studies, it is lower than the performance in our study. Also other SoTA models could have been applied in the study
Benbrahim et al. [37]	Chest X-ray	160 COVID-19 (+) 160 Normal	InceptionV3 and ResNet50	–	99	Only binary classification was made. The classification performance is good. However, the data set is considerably small
Al-antari et al. [61]	Chest X-ray	326 COVID-19 120 Pneumonia 866 other seven lung diseases	YOLO	YOLO detection algorithm	90.67	Detection was made using the YOLO algorithm. Unlike our study, no cascade structure fed by YOLO was used. The classification performance is low
Nigam et al. [62]	Chest X-ray	5634 COVID-19 6000 Normal 5000 Others	VGG16, DenseNet121, Xception, NASNet and EfficientNet	YOLO detection algorithm	93.48	A cascade structure fed by YOLO was used as a method as in our study. However, the classification performance is lower. The impact of YOLO could not be revealed as the performances of models not fed with YOLO were not included in the study
Aslan et al. [30]	Chest X-ray	1095 COVID-19 (+) 1345 Pneumonia 1341 Normal	Hybrid CNN based on AlexNet	ANN-based automatic lung segmentation	98.70	ANN-based chest zone segmentation was performed, and the images were given as input to the AlexNet-based hybrid CNN. It is similar to our study as a cascade model was used. The classification performance is slightly higher than our study. However, data augmentation was carried out in the study. Data augmentation is not regarded as a good practice in medical imaging

Table 4 (continued)

Study	Type of images	Number of cases	Method used	Segmentation or location attention	Accuracy (%)	Remarks
Hira et al.[31]	Chest X-ray	224 COVID-19 (+) 700 Pneumonia 504 Normal	Se-ResNeXt-50	–	97.55	Classification was carried out with nine pre-trained CNN models (AlexNet, GoogleNet, ResNet-50, Se-ResNet-50, DenseNet121, Inception V4, Inception ResNet V2, ResNeXt-50, and Se-ResNeXt-50). The highest classification performance was attained with Se-ResNeXt-50. The classification performance is slightly higher than our study
Butt et al. [38]	Chest CT	219 COVID-19 (+) 224 Pneumonia 175 Healthy people	RESNET	–	86.7	The data sets used in these studies consist of CT images, and the classification performances are considerably lower than our study
Ying et al. [39]	Chest CT	777 COVID-19 (+) 708 Healthy	DRE-NET	–	94	
Zheng et al. [40]	Chest CT	313 COVID-19 (+) 229 COVID-19 (−)	DECOVNET	-	90.1	
Xu et al. [63]	Chest CT	219 COVID-19 (+) 224 Pneumonia 175 Healthy	ResNet-18 + location based attention	Image patch vote and Noisy-OR Bayesian function	86.7	
Proposed study	Chest X-ray	500 No-Findings 125 COVID-19 (+)	Cascade VGGCOV19-NET		**99.84**	
		500 No-Findings 500 Pneumonia 125 COVID-19 (+)			**97.16**	

Bold values indicate the results obtained in this study

There are also studies [38–40, 63, 64] diagnosing COVID-19 using chest CT in the literature. One of these studies [64] states that mobile devices have limited storage and calculating capacity and suggests an architecture including robot, edge, and cloud layers so that deep learning applications can be run on mobile devices. They detected COVID-19 from CT images using DenseNet169 on this architecture. The accuracy values in these studies are relatively lower than those of studies conducted using X-rays. As specified by Ozturk et al. [9], CT is very costly and not easily accessible because scanners are only available in major health centers. When compared to X-rays, CT causes higher radiation amounts in the patients. Consequently, we recommend using a deep learning model with X-ray images because they are more accessible than CT with lower radiation dosages.

Because X-ray radiographies are easily accessible, they are commonly used for the diagnosis of COVID-19 in health centers worldwide. The Cascade VGGCOV19-NET model can be used as auxiliary tool in the diagnosis of COVID-19 in a much shorter period than X-ray radiographies. The main advantages of this model are as follows:

- They classify chest X-ray images that can be very easily, rapidly, and smoothly attained from any hospital without needing extra feature extraction methods.
- They generally have higher classification performance than the other studies in the literature.
- As in some studies in the literature, the original state-of-the-art CNN architecture was not used. VGG19 CNN architecture has been modified and its classification performance increased.
- New Cascade VGGCOV19-NET model based on the YOLOv3 detection algorithm and the VGG19 pre-trained CNN architecture has increased the classification performance and guided future studies.
- They are able to correctly distinguish COVID-19 at higher ratios than pneumonia and no-findings.

Cascade VGGCOV19-NET detects COVID-19 in minutes and at a high accuracy rate. For this reason, it can be used in areas where there is a shortage of specialist doctors and radiologists. In addition, even in small healthcare facilities where only an X-ray device is available, patients can be diagnosed using the model and those who are positive for COVID referred to more advanced health centers for immediate treatment. Another advantage of the model in application is that it prevents patients who are detected negatively from entering PCR tests. Thus, unnecessary clutter in health centers can be avoided. The trained Cascade VGGCOV19-NET model is saved in the computer environment. After a simple interface is designed for the model, the X-ray image can be easily loaded into the system and results obtained very quickly. Also, X-ray images can be rapidly transferred to the physician's computer without the need for additional procedures. All the physician need do is upload the X-ray image to the system with a few mouse clicks to see the result. From this point of view, Cascade VGGCOV19-NET can easily be used for pre-diagnosis.

6 Conclusion

In this study, the deep CNN model modified VGG19 architecture-based VGGCOV19-NET was presented to diagnose COVID-19 cases from chest X-ray images. In addition, we created the Cascade VGGCOV19-NET model by feeding this model with the YOLOv3 algorithm. The VGGCOV19-NET model performs the binary class and multi-class classification tasks with an accuracy of 98.72% and 88.89%, respectively. These accuracy values are 99.84% and 97.16%, respectively, in the Cascade VGGCOV19-NET model. Feeding VGGCOV19-NET with the YOLOv3 algorithm increases the classification performance. Moreover, it decreases the training and test time by approximately 70% in the binary classification and by 60% in the multi-class classification. The Cascade VGGCOV19-NET is an important and original model that detects COVID-19 with high accuracy that will lead to future studies. This model automatically detects COVID-19 with high accuracy from X-ray images, which are widely used and easily accessible for the diagnosis of COVID-19 in health centers worldwide. Conventional pneumonia has great similarities with COVID-19, and there is a severe shortage of specialists. For this reason, Cascade VGGCOV19-NET, an artificial intelligence-supported automatic detection system, is especially important in detecting COVID-19. The Cascade VGGCOV19-NET model is a tool that can assist doctors by diagnosing COVID-19 rapidly and easily at low cost. In addition, the YOLO-supported COVID-19 detection method proposed in this study can be applied to various fields, such as tumor detection from brain images, skin cancer detection, and implant-type detection from implant images.

One limitation of the study is the use of a limited number of COVID-19 X-ray images. As medical images for COVID are obtained from local hospitals, it is very difficult to obtain various data sets for such studies conducted with CNN. The number of data sets in the literature was not enough when we initiated this study. Furthermore, we could not use data augmentation as it is not regarded as a good practice in medical imaging. Another limitation of the study is that the proposed cascade method was used only on X-ray images. On the other hand, it is thought that the proposed cascade method could successfully classify CT images. In a future study, we will try to detect COVID-19 through different CNN models on a very large mixed data set by adding other open-source X-ray and computed tomography images in the literature to the X-ray images data set used in this study. Another limitation of the study is that the chest zone was determined only as a quadrant with the YOLO algorithm. In particular, the three-class classification performance can increase if the right and left lung zones are segmented using image processing techniques. This segmentation will be carried out as a future study, and the results will be compared with the YOLO-based model. Moreover, another limitation of the study is that only the VGG19 SoTA (state-of-the-art) model was used. Therefore, SoTA models such as EfficientNet, ResNet, MobileNet, Densenet169, and Inceptionv3 fed with YOLO can be used to improve the performance of the COVID diagnosis system.

Funding No funds, grants, or other support was received.

Declarations

Conflicts of interest The author declare that there is no any conflict of interest.

References

1. Zu ZY, Di Jiang M, Xu PP et al (2020) Coronavirus disease 2019 (COVID-19): a perspective from China. Radiology 296:E15–E25
2. Wu F, Zhao S, Yu B et al (2020) A new coronavirus associated with human respiratory disease in China. Nature 579:265–269. https://doi.org/10.1038/s41586-020-2008-3
3. D'Angelo G, Palmieri F (2020) Discovering genomic patterns in SARS-CoV-2 variants. Int J Intell Syst 35:1680–1698. https://doi.org/10.1002/int.22268
4. Wu Z, McGoogan JM (2020) Characteristics of and important lessons from the coronavirus disease 2019 (COVID-19) outbreak in China. JAMA 323:1239. https://doi.org/10.1001/jama.2020.2648
5. Garain A, Basu A, Giampaolo F et al (2021) Detection of COVID-19 from CT scan images: a spiking neural network-based approach. Neural Comput Appl. https://doi.org/10.1007/s00521-021-05910-1
6. Batiha O, Al-Deeb T, Al-zoubi E, Alsharu E (2020) Impact of COVID-19 and other viruses on reproductive health. Andrologia 52:e13791. https://doi.org/10.1111/and.13791
7. Harmon SA, Sanford TH, Xu S et al (2020) Artificial intelligence for the detection of COVID-19 pneumonia on chest CT using multinational datasets. Nat Commun 11:4080. https://doi.org/10.1038/s41467-020-17971-2
8. Mahase E (2020) Coronavirus: covid-19 has killed more people than SARS and MERS combined, despite lower case fatality rate. BMJ 368:m641. https://doi.org/10.1136/bmj.m641
9. Ozturk T, Talo M, Yildirim EA et al (2020) Automated detection of COVID-19 cases using deep neural networks with X-ray images. Comput Biol Med 121:103792. https://doi.org/10.1016/j.compbiomed.2020.103792
10. Waheed A, Goyal M, Gupta D et al (2020) CovidGAN: data augmentation using auxiliary classifier GAN for improved Covid-19 detection. IEEE Access. https://doi.org/10.1109/ACCESS.2020.2994762
11. Khan AI, Shah JL, Bhat MM (2020) CoroNet: a deep neural network for detection and diagnosis of COVID-19 from chest X-ray images. Comput Methods Programs Biomed 196:105581. https://doi.org/10.1016/j.cmpb.2020.105581
12. Singhal T (2020) A review of coronavirus disease-2019 (COVID-19). Indian J Pediatr 87:281–286. https://doi.org/10.1007/s12098-020-03263-6
13. Lee EYP, Ng M-Y, Khong P-L (2020) COVID-19 pneumonia: What has CT taught us? Lancet Infect Dis 20:384–385. https://doi.org/10.1016/S1473-3099(20)30134-1
14. Chan JF-W, Yuan S, Kok K-H et al (2020) A familial cluster of pneumonia associated with the 2019 novel coronavirus indicating person-to-person transmission: a study of a family cluster. Lancet 395:514–523. https://doi.org/10.1016/S0140-6736(20)30154-9
15. Han D, Liu Q, Fan W (2018) A new image classification method using CNN transfer learning and web data augmentation. Expert Syst Appl 95:43–56. https://doi.org/10.1016/j.eswa.2017.11.028
16. Hemdan EE-D, Shouman MA, Karar ME (2020) COVIDX-Net: a framework of deep learning classifiers to diagnose COVID-19 in X-ray images. https://arxiv.org/abs/2003.11055
17. Ouchicha C, Ammor O, Meknassi M (2020) CVDNet: a novel deep learning architecture for detection of coronavirus (Covid-19) from chest X-ray images. Chaos Solitons Fractals 140:110245. https://doi.org/10.1016/j.chaos.2020.110245
18. Haghanifar A, Majdabadi MM, Choi Y et al (2020) COVID-CXNet: detecting COVID-19 in frontal chest X-ray images using deep learning. https://arxiv.org/abs/2006.13807
19. Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22:1345–1359. https://doi.org/10.1109/TKDE.2009.191
20. Weiss K, Khoshgoftaar TM, Wang D (2016) A survey of transfer learning. J Big Data 3:9. https://doi.org/10.1186/s40537-016-0043-6
21. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. https://arxiv.org/abs/1409.1556
22. Sukegawa S, Yoshii K, Hara T et al (2020) Deep neural networks for dental implant system classification. Biomolecules 10:984. https://doi.org/10.3390/biom10070984
23. Wan X, Zhang X, Liu L (2021) An improved VGG19 transfer learning strip steel surface defect recognition deep neural network based on few samples and imbalanced datasets. Appl Sci 11:2606. https://doi.org/10.3390/app11062606
24. Arshad H, Khan MA, Sharif MI et al (2020) A multilevel paradigm for deep convolutional neural network features selection with an application to human gait recognition. Expert Syst. https://doi.org/10.1111/exsy.12541
25. Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement. https://arxiv.org/abs/1804.02767
26. Punn NS, Sonbhadra SK, Agarwal S, Rai G (2020) Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and deepsort techniques. https://arxiv.org/abs/2005.01385
27. Ge L, Dan D, Li H (2020) An accurate and robust monitoring method of full-bridge traffic load distribution based on YOLO-v3 machine vision. Struct Control Heal Monit. https://doi.org/10.1002/stc.2636
28. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 779–788
29. Wang L, Lin ZQ, Wong A (2020) COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. Sci Rep 10:19549. https://doi.org/10.1038/s41598-020-76550-z
30. Aslan MF, Unlersen MF, Sabanci K, Durdu A (2021) CNN-based transfer learning–BiLSTM network: a novel approach for COVID-19 infection detection. Appl Soft Comput 98:106912. https://doi.org/10.1016/j.asoc.2020.106912
31. Hira S, Bai A, Hira S (2021) An automatic approach based on CNN architecture to detect Covid-19 disease from chest X-ray images. Appl Intell 51:2864–2889. https://doi.org/10.1007/s10489-020-02010-w
32. Medhi K, Jamil M, Hussain MI (2020) Automatic detection of COVID-19 Infection from chest X-ray using deep learning. medrxiv. https://doi.org/10.1101/2020.05.10.20097063
33. Harit A, Shubharthi D, Bagish C (2020) Performance result for detection of COVID-19 using deep learning. Int J Innov Technol Explor Eng 9:699–703

34. Ahammed K, Satu MS, Abedin MZ et al (2020) Early detection of coronavirus cases using chest X-ray images employing machine learning and deep learning approaches. medRxiv. https://doi.org/10.1101/2020.06.07.20124594
35. Apostolopoulos ID, Mpesiana TA (2020) Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. Phys Eng Sci Med 43:635–640. https://doi.org/10.1007/s13246-020-00865-4
36. Narin A, Kaya C, Pamuk Z (2021) Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. Pattern Anal Appl 24:1207–1220. https://doi.org/10.1007/s10044-021-00984-y
37. Benbrahim H, Hachimi H, Amine A (2020) Deep transfer learning with apache spark to detect COVID-19 in chest X-ray images. Rom J Inf Sci Technol 23:117–129
38. Butt C, Gill J, Chun D, Babu BA (2020) RETRACTED ARTICLE: deep learning system to screen coronavirus disease 2019 pneumonia. Appl Intell 6:1122–1129. https://doi.org/10.1007/s10489-020-01714-3
39. Ying S, Zheng S, Li L et al (2020) Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images. medRxiv. https://doi.org/10.1101/2020.02.23.20026930
40. Zheng C, Deng X, Fu Q et al (2020) Deep learning-based detection for COVID-19 from chest CT using weak label. medRxiv. https://doi.org/10.1101/2020.03.12.20027185
41. Cohen JP, Morrison P, Dao L (2020) COVID-19 image data collection. https://arxiv.org/abs/2003.11597
42. Wang X, Peng Y, Lu L et al (2017) ChestX-Ray8: hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 3462–3471
43. Ganapathy N, Veeranki YR, Swaminathan R (2020) Convolutional neural network based emotion classification using electrodermal activity signals and time-frequency features. Expert Syst Appl 159:113571. https://doi.org/10.1016/j.eswa.2020.113571
44. Hoseinzade E, Haratizadeh S (2019) CNNpred: CNN-based stock market prediction using a diverse set of variables. Expert Syst Appl 129:273–285. https://doi.org/10.1016/j.eswa.2019.03.029
45. Han D, Liu Q, Fan W (2018) A new image classification method using CNN transfer learning and web data augmentation. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2017.11.028
46. Seo Y, Shin K (2019) Hierarchical convolutional neural networks for fashion image classification. Expert Syst Appl 116:328–339. https://doi.org/10.1016/j.eswa.2018.09.022
47. Yamashita R, Nishio M, Do RKG, Togashi K (2018) Convolutional neural networks: an overview and application in radiology. Insights Imaging 9:611–629. https://doi.org/10.1007/s13244-018-0639-9
48. Scherer D, Müller A, Behnke S (2010) Evaluation of pooling operations in convolutional architectures for object recognition. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). pp 92–101
49. Toğaçar M, Ergen B, Cömert Z, Özyurt F (2020) A deep feature learning model for pneumonia detection applying a combination of mRMR feature selection and machine learning models. IRBM 41:212–222. https://doi.org/10.1016/j.irbm.2019.10.006
50. Kuo C-CJ (2016) Understanding convolutional neural networks with a mathematical model. J Vis Commun Image Represent 41:406–413. https://doi.org/10.1016/j.jvcir.2016.11.003
51. O'Shea K, Nash R (2015) An introduction to convolutional neural networks. https://arxiv.org/abs/1511.08458
52. Boureau YL, Ponce J, Lecun Y (2010) A theoretical analysis of feature pooling in visual recognition. In: ICML 2010—proceedings, 27th international conference on machine learning. Haifa, Israel
53. Mahmud T, Rahman MA, Fattah SA (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. Comput Biol Med 122:103869. https://doi.org/10.1016/j.compbiomed.2020.103869
54. Aktaş A, Doğan B, Demir Ö (2020) Tactile paving surface detection with deep learning methods. J Fac Eng Archit Gazi Univ 35:1685–1700. https://doi.org/10.17341/gazimmfd.652101
55. Loey M, Manogaran G, Taha MHN, Khalifa NEM (2021) Fighting against COVID-19: a novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection. Sustain Cities Soc 65:102600. https://doi.org/10.1016/j.scs.2020.102600
56. Ahamad MM, Aktar S, Rashed-Al-Mahfuz M et al (2020) A machine learning model to identify early stage symptoms of SARS-Cov-2 infected patients. Expert Syst Appl 160:113661. https://doi.org/10.1016/j.eswa.2020.113661
57. Altman DG, Bland JM (1994) Statistics notes: diagnostic tests 1: sensitivity and specificity. BMJ 308:1552–1552. https://doi.org/10.1136/bmj.308.6943.1552
58. Davis J, Goadrich M (2006) The relationship between precision-recall and ROC curves. In: Proceedings of the 23rd international conference on machine learning—ICML'06. ACM Press, New York, pp 233–240
59. Janssens ACJW, Martens FK (2020) Reflection on modern methods: revisiting the area under the ROC Curve. Int J Epidemiol 49:1397–1403. https://doi.org/10.1093/ije/dyz274
60. Chowdhury MEH, Rahman T, Khandakar A et al (2020) Can AI help in screening Viral and COVID-19 pneumonia? IEEE Access 8:132665–132676. https://doi.org/10.1109/access.2020.3010287
61. Al-antari MA, Hua C-H, Lee S (2020) Fast deep learning computer-aided diagnosis against the novel COVID-19 pandemic from digital chest X-ray images. Appl Intell. https://doi.org/10.21203/rs.3.rs-36353/v1
62. Nigam B, Nigam A, Jain R et al (2021) COVID-19: automatic detection from X-ray images by utilizing deep learning methods. Expert Syst Appl 176:114883. https://doi.org/10.1016/j.eswa.2021.114883
63. Xu X, Jiang X, Ma C et al (2020) A Deep learning system to screen novel coronavirus disease 2019 pneumonia. Engineering 6:1122–1129. https://doi.org/10.1016/j.eng.2020.04.010
64. Mo H, Ding S, Yang S et al (2020) The role of edge robotics as-a-service in monitoring COVID-19 infection. https://arxiv.org/abs/2011.08482