Neural Computing and Applications (2024) 36:5347–5365
https://doi.org/10.1007/s00521-023-09346-7

ORIGINAL ARTICLE

# COVID-19 classification based on a deep learning and machine learning fusion technique using chest CT images

Gerges M. Salama[1] · Asmaa Mohamed[1] · Mahmoud Khaled Abd-Ellah[2]

Received: 14 March 2023 / Accepted: 26 November 2023 / Published online: 29 December 2023

## Abstract

Coronavirus disease (COVID-19), impacted by SARS-CoV-2, is one of the greatest challenges of the twenty-first century. COVID-19 broke out in the world over the last 2 years and has caused many injuries and killed persons. Computer-aided diagnosis has become a necessary tool to prevent the spreading of this virus. Detecting COVID-19 at an early stage is essential to reduce the mortality risk of patients. Researchers seek to find rapid solutions based on techniques of Machine Learning and Deep Learning. In this paper, we introduced a hybrid model for COVID-19 detection based on machine learning and deep learning models. We used 10 different deep CNN network models to extract features from CT images. We extract features from different layers in each network and find the optimum layer that gives the best-extracted features for each CNN network. Then, for classifying these features, we used five different classifiers based on machine learning. The dataset consists of 2481 CT images divided into COVID-19 and non-COVID-19 categories. Three folds are extracted with a different size between testing and training. Through experiments, we define the best layer for all used CNN networks, the best network, and the best-used classifier. The measured performance shows the superiority of the proposed system over the literature with a highest accuracy of 99.39%. Our models are tested with the three folds that gained maximum average accuracy. The result is 98.69%.

**Keywords** COVID-19 · CT images · CNN · Machine learning · Deep learning · SVM

## 1 Introduction

Coronavirus (COVID-19) disease is a global health crisis that occurred in the last 2 years and caused many deaths in many countries around the world. Some symptoms of COVID-19 are variable based on the kind of variant contracted, ranging from slight symptoms to critical and deadly illness [1]. Normal side effects incorporate coughing, fever, and loss of smell and taste, and more uncommon ones include migraines, nasal blockage and runny nose, sore throat, muscle torment, loose bowels, eye bothering, toes swelling or turning purple, and moderate to serious breathing troubles. In the last 2 years, the main challenge has been the early detection of COVID-19 to avoid an increase in disease and death in all countries around the world [2]. Hence, much research is introduced since the beginning spread of COVID-19, which is based on different models of machine learning and deep learning for the detection, classification, and segmentation of this disease [3].

Nowadays, machine learning (ML) and deep learning (DL) have become the main core of artificial intelligence. Machine learning (ML) and deep learning (DL) have become increasingly important in various fields due to their ability to analyze large amounts of data and extract patterns and insights that can inform decision-making. Different fields of this exist, such as healthcare for the diagnosis of different diseases, the energy sector for the modeling of fluidized adsorption beds [4], chemistry for predicting the NOx emissions from the coal combustion into circulating fluidized-bed combustors [5], or power generation in the industry sector [6]. In healthcare, many ML and DL models are developed in the medical area [7]. A convolutional

✉ Mahmoud Khaled Abd-Ellah
Mahmoud-khaled@eru.edu.eg

[1] Department of Electrical Engineering, Faculty of Engineering, Minia University, Minia 61111, Egypt

[2] Faculty of Artificial Intelligence, Egyptian Russian University, Cairo 11829, Egypt

neural network (CNN) is considered a special sort of deep learning architecture based on neural networks and consists of three main layers (convolutional layer, pooling layer, and fully connected layer) [8]. CNN is appropriate for several fields of computer vision. Therefore, a higher performance of the convolutional neural network (CNN) with images could help many researchers in the diagnosis of COVID-19 [9].

Different types of medical images can be used in the diagnosis of any disease, such as X-ray, MRI, CT scan, PET scan, and ultrasound [10]. All these images can be used in diagnostic tests for COVID-19. The most common types are the chest X-ray (CXR) and computed tomography (CT) scan. However, CT is the essential radiographic test to assess COVID-19 as it is accurate, has higher misdiagnosis rates, and is quicker [11].

Imaging classification for COVID-19 has emerged as an essential tool for diagnosing and monitoring disease and other lung diseases. Different radiological imaging techniques such as chest X-rays and CT scans can reveal characteristic features of the disease [12]. These extracted features can help differentiate COVID-19 from other respiratory illnesses and provide valuable information on the severity and progression of the disease. Imaging classification also plays a crucial role in identifying patients who are at high risk of developing severe illness, such as those with pre-existing lung disease or immunocompromised individuals. Additionally, imaging can aid in the early detection of complications such as pneumonia, which can be life-threatening if left untreated. Imaging classification has also been used to identify imaging biomarkers that may predict disease severity and clinical outcomes [13]. For example, studies have shown that the extent of lung involvement on CT scans is associated with the likelihood of ICU admission and mortality. Overall, imaging classification plays a critical role in the diagnosis, management, and prognosis of COVID-19, making it an indispensable tool for healthcare providers. Overall, imaging classification for COVID-19 is an essential component of the diagnostic process, allowing for early detection, appropriate treatment, and improved patient outcomes. The main problem is to get more accurate results in the classification of COVID-19. Therefore, our objective is to define the most accurate ML classifier with the highest-performance CNN model to extract features of the network. In this research paper, we can address some popular deep learning models based on convolutional neural network models using different machine learning classifiers [14]. There are 10 deep convolutional networks: Xception [15], Darknet53 [16], Vgg19 [17], Alexnet [18], Googlenet [19], Mobilenetv2 [20], Squeezenet [21], Darknet19 [22], Resnet50 [21], and Resnet101 [21]). These are used to extract features from our dataset which consists of 2,482 CT images. We focused on applying different machine learning classifiers for binary classification (COVID-19 and non-COVID-19) using extracted features from different deep convolutional neural networks. In each model, we test random layers to define the best layer to achieve high accuracy and define the best results for network model with the best used classifier with the three different database folds.

There are some key contributions of this paper that can be summarized as follows:

1) Extract the most relevant features which fit the highest accuracy and categorize into two levels: the network level and the layer level.
2) Find the best layer in the 10 selected deep CNN networks for feature extraction.
3) Find the best classifier from the selected ML classifiers which could give higher performance.
4) Proposed hybrid models of both machine learning and deep learning can give us higher results than each separately.
5) Present the highest 10 models from the proposed hybrid models for COVID-19 detection.
6) Present the best models for COVID-19 detection among all the proposed models and the literature.

The paper is organized as follows. Section 2 presents some related work and how this could be used to diagnose COVID-19. Subsequently, details about our proposed methodology and information about some used classifiers are described in Sect. 3. Section 4 introduces details about our experimental setup and the dataset. Section 5 introduces the obtained results and a detailed discussion about the obtained results. Finally, the conclusion is presented in Sect. 6.

## 2 Related work

Many effective works that are based on machine learning and deep learning have been introduced using chest radiography images such as X-rays images and CT scans through the last 2 years from the start of the COVID-19 crisis. In this section, we can discuss the most relevant works that are based on different machine learning classifiers and different deep learning models to extract features from the used dataset images. Therefore, we will highlight their success and accomplished results.

In this paper [23], a deep method to extract CXR image features by using Convolutional Neural Networks (CNNs) called Alexnet is presented. The study classified images to recognize COVID-19 disease by Support Vector Machine (SVM) using nonlinear kernel-function algorithms. The researchers used 146 CXR images which were divided into two classes, COVID-19 and non-COVID-19 (healthy). They categorized the results as follows: Accuracy (91.53%), Recall (91.68%), False Positive Rate (91.06%), and True Negative Rate (8.32%). Researchers in [24] introduced a new deep model called as COVIDX-Net which included seven deep CNN models: VGG19, MobileNet, Densenet121, InceptionV3, ResNetV2, Inception-ResNet-V2, and Xception. Each model could extract deep features from X-ray images to classify non-COVID-19 and COVID-19 cases. They used a dataset consisting of 50 X-ray images and divided it into two classes. In their evaluation process, they used 80% of the images for training and 20% for testing. Their model could achieve the best f1-score for Densenet121 with a value of 0.91, and the same accuracy values of 90% for Vgg19 and Densent121.

In [25], the researchers' model was based on using 11 different deep CNN models to extract features from X-ray images plus the methodology of support vector machine (SVM) as a classifier which could categorize images into three categories: non-COVID-19, COVID-19, and pneumonia. Their dataset consisted of 381 CXR images which were divided into 127 confirmed COVID-19 images, 127 confirmed pneumonia images, and 127 healthy images. Resnet50 plus SVM could achieve the best results among other models as they achieved accuracy, sensitivity, FPR, and F1 score values of 95.33%, 95.33%, 2.33%, and 95.34% respectively. Researchers in [27] implemented different algorithms for the diagnosis of lung diseases like COVID-19 and pneumonia using GLCM and HOG features to extract features from chest X-rays (CXR). They used 6300 patches of lung images. They used Support Vector Machine (SVM) and Artificial Neural Networks (ANN) as classifiers. They achieved a maximum accuracy value of 93.73% for SVM.

In [27], deep-learning-based methods were introduced, namely deep learning feature extraction, fine-tuning based on some pre-trained convolutional neural networks, and end-to-end training of a developed CNN model. The researchers used a dataset consisting of 180 COVID-19 and 200 healthy chest X-ray images. They used some pre-trained deep CNN models: ResNet18, ResNet50, ResNet101, VGG16, and VGG19 to extract features. They used a Support Vector Machines (SVM) classifier with different Kernel functions to classify the extracted features from used CNN models. They found that ResNet50 with Linear Kernel SVM classifier could achieve the best accuracy value of 94.7% between other used models. In this work [28], they proposed a deep-learning CNN architecture consisting of 15 layers to extract features from their collected CT images. They collected 10 cases of COVID-19 pneumonia chest CT images from RadioPaedia for the classification of the COVID-19 pneumonia infection as normal. They used two main layers to extract deep features, the global average pool, and fully connected layers. These layers are combined using the approach of max-layer detail (MLD). Later, a Correntropy technique is embedded in the main design to choose the most discriminant features from the pool of features. For the final classification, they used a one-class kernel extreme learning machine classifier which could achieve an average accuracy of 95.1%, and 95.1% sensitivity, 95% specificity and 94% precision rates.

In another study [29], researchers used a dataset consisting of 15,153 X-ray images and classified these into three classes: viral pneumonia, normal, and COVID-19. They processed this dataset and then used Local Binary Model to extract features from it. They used six classifiers, Cubic SVM, Linear Discriminant (LD), Quadratic Discriminant (QD), Ensemble, Kernel NaiveBayes (KNB), and KNN Weighted, to classify input images. Their model achieved the highest performance by using Cubic SVM with an accuracy value of 98.05%. In [30], researchers introduced a detection and classification system for COVID-19 using X-ray images. They introduced six models beginning with image processing, segmentation, feature extraction, and two classifiers: KNN and SVM. These six models are named HOG-KNN, LBP-KNN, Haralick-KNN, HOG-SVM, LBP-SVM, and Haralick-SVM. They used 5,000 X-ray images for testing five folds validation. The LBP-KNN model outperformed the other five models with an average accuracy of 98.66%, 97.76% sensitivity, 100% specificity, and 100% precision.

In [31], researchers presented a classification algorithm for COVID-19 cases using lung CT scans. They used the q-transform model to extract features from 276 CT scans. Then, they used three ML classifiers: "k-nearest neighbor", "decision tree", and SVM. They achieved the highest performance with the SVM classifier with 98.25% accuracy, 95.30% sensitivity, and 97.60% specificity. In [32], researchers used six CNN architectures: VGG16, DeseNet121, MobileNet, NASNet, Xception, and EfficientNet. They used 3873 CT images for binary classification (COVID and non-COVID). They achieved the highest performance with VGG16, which had an accuracy of 97.68% compared to other used architectures. In this paper [33], researchers described a CNN model, which employed three image classes: normal, COVID-19, and viral pneumonia. These images were divided as follows: 219 for COVID-19, 1341 for normal images, and 1345 for viral pneumonia cases. The CNN model had an accuracy of around 94% on the test data for three classification classes, which contained only 32 images from all classes. Its recall and precision values for COVID-19 were 94% and 95%, respectively, whereas the VGG16 model achieved 97% accuracy and the VGG19 model achieved 97% accuracy for the three classes' classification.

In [34], researchers suggested a new model of CNN called ResF which was comprised of five major blocks. Each block had two succeeding elements: a ResF module and a connector ResF module. Their dataset was based on using freely available LUS images and video datasets. They used 121 videos divided into 45 for COVID-19, 23 for bacterial pneumonia, and 53 for healthy people. There were also 40 images: COVID-19 (18), bacterial pneumonia (7), and healthy (15). The proposed system could achieve 92.5% precision and 91.8% accuracy value. Mohammad Rahimzadeh et al. [35] proposed some training model techniques that helped the network's learning when faced with an unbalanced dataset. In addition, they proposed a new model which was a hybrid of Xception and ResNet50V2 networks. In this work, they used two open-source datasets. The first database was X-ray images, which were classified into 4 classes: 180 images for COVID-19 cases and 42 images for streptococcus, pneumocystis, and SARS, which were classed as pneumonia. The second database was obtained from the Kaggle database which contained 6012 pneumonia cases and 8851 normal cases. Finally, they tested their network on 11302 images and proposed an average accuracy value for detecting only COVID-19 cases at 99.50%, where the average accuracy for all classes was 91.4%.

## 3 Methodology

Convolutional Neural Networks or CNN can have a significant role in image classification. Its technique is based on using several small filters to define the features in each layer. CNN consists of some sequenced layers, where each layer can extract some features from input images. The main layers of CNN are the convolutional layer, sampling layer, activation layer, and fully connected layers. Deep feature extraction from chest X-ray images is based on extracting features based on some pre-trained CNN models.

Our hybrid model consists of four stages as shown in Fig. 1. The process begins by inputting our dataset of CT images. Then, some preprocessing techniques are applied to the used images, such as resizing and augmentation. Different CNN architectures—Xception, Darknet53, Vgg19, Alexnet, Googlenet, Mobilenetv2, Squeezenet, Darknet19, Resnet50, and Resnet101—are used to extract features from this dataset. Finally, five machine learning classifiers are used to classify the dataset and select the best classifier that can give our model more performance.

### 3.1 Preprocessing stage

Pre-processing for the input dataset is a critical step to obtain more adjustable and fitting images for the classification process. Firstly, we start resizing these images to be compatible with the used CNN network. All images are resized to fit with each selected network. The augmentation process is also applied to input images in which we can increase the size of the dataset without the need to acquire new images [36].

### 3.2 Feature extraction stage

The deep feature extraction process can help the classifier to identify the image with higher accuracy. Different CNN models are applied for feature extraction. To extract features from the used dataset, we selected 10 CNN models, since CNN networks have the robust ability to extract complex and deep features that convey each image in more detail [37]. Xception consists of 170 layers that involve Depthwise Separable Convolutions. It also has 36 convolutional layers [15]. Darknet53 has 184 layers and is based on using residual connections [16]. VGG19 has 47 layers [17]. Figure 2 illustrates the extracted features at layer 33 (Relu Layer).

Alexnet was proposed in 2012 and consisted of 25 layers in total [18]. GoogLeNet is based on the inception architecture and includes 144 layers. It used the concept of inception modules, in which the network can choose between various sizes of filters in each block [19]. MobileNetV2 has 154 layers and is based on an upturned residual structure in which the residual connections are linked between the bottleneck layers [20]. Squeezenet is a type of convolutional neural network with 68 layers [21]. Darknet19 involves 64 layers [22]. ResNet50 has 177 layers based on CNN architecture [21]. ResNet101 involves 347 layers based on CNN architecture with an input image size of 224 × 224 × 3 [21]. Every network layer of the CNN models' performance is evaluated. In this phase, the ideal number of network layers to use for the fastest feature extraction and best accuracy will be determined. We provide the best 4 layers for feature extraction from these networks to define the best layer with higher performance. The number of extracted features in each layer for these models is shown in Table 1.

### 3.3 classification stage

Binary classification (COVID and non-COVID) is carried out after the stage of feature extraction in each one of the selected layers of the used CNN network. In each model, we used some selected layers with five used machine

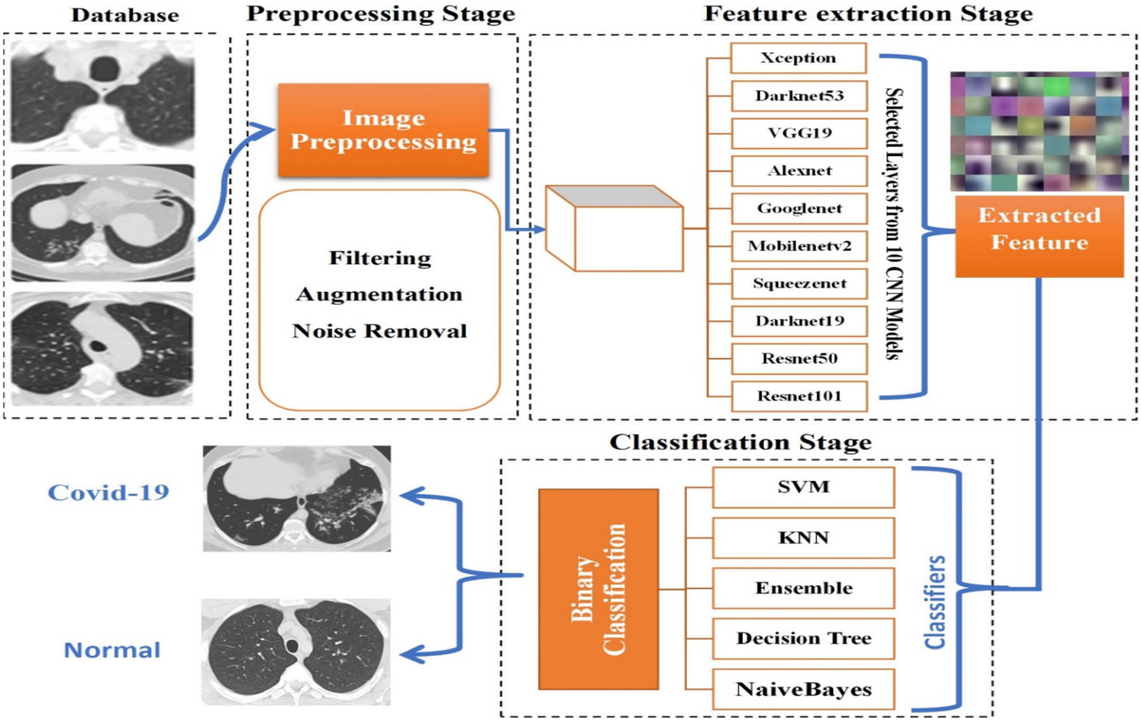


Fig. 1 Our hybrid classification model based on machine and deep learning

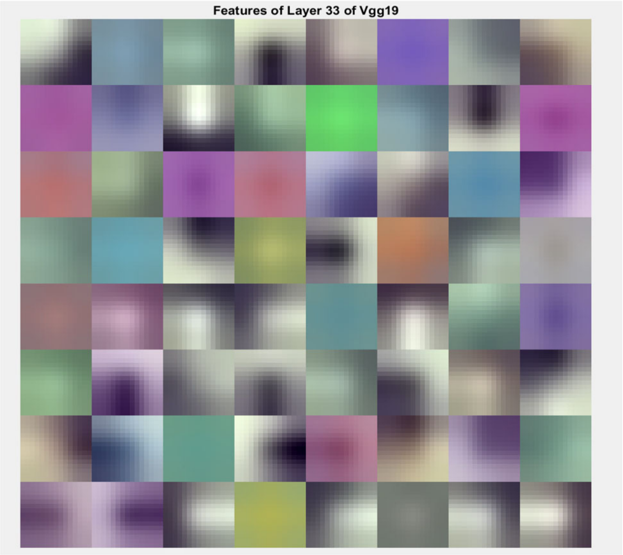


Fig. 2 Extracted features of layer 33 of VGG19

learning classifiers: support Vector Machine (SVM), *K*-nearest neighbors (KNN), Ensemble Classifier NaiveBayes Classifier, and Decision Tree (DT) Classifier.

Support Vector Machine (SVM) is a classical kernel learning classifier that seeks to get the optimal separating hyperplane among different classes by focusing on the training cases named as support vectors. It can be placed at the edge of class descriptors [38]. The hyperplane is chosen in such a way that it maximizes the margin, which is the distance between the hyperplane and the closest data points of each class (support vectors). SVM can build this optimal separating hyperplane based on a kernel function (K). All images whose feature vectors lie on one side of the separating hyperplane belong to class 1 and the others belong to class -1 [39]. Kernel is a mathematical function that can be used in SVM. Even if the data points are not linearly separable in the input space, SVM can be used to map the original data points onto high-dimensional feature spaces, making it simple to identify the hyperplane [40]. There are different types of SVM. Our work is based on error correcting output codes and the Support Vector Machine (ECOC-SVM). The ECOC approach is used to correct the data error when inputted into a channel. It converts the figure from binary classification to a multiclass classification [41]. ECOC-SV is used to solve the problems of online identification and feature extraction. Figure 3 is a typical structure of the ECOC-SVM multi-class classifier. By applying ECOC-SVMs to multi-classification, it transforms the *M* classes classification to N classes classification by using ECOC matrix. ECOC-SVM functions according to the following steps:

- In training, by SVM binary partition, classes are divided into two subsets (0 or 1) for each row in the coding matrix. The row containing a value of 0 is designated as the first class, while the row with a value of 1 is considered the second class. Each column of the matrix is assigned a codeword corresponding to its respective

**Table 1** Extracted features from different layers using some deep feature extraction CNN models2

| Model | Layers | Layer name | Number of extracted features |
|---|---|---|---|
| Xception | **Layer 79** | Convolution | 262808 |
| | **Layer 108** | Relu | 262808 |
| | **Layer 117** | Grouped convolution | 262808 |
| | **Layer 134** | Relu | 262808 |
| DarkNet53 | **Layer 32** | Convolution | 262144 |
| | **Layer 34** | Leaky relu | 262144 |
| | **Layer 42** | Convolution | 131072 |
| | **Layer 182** | Convolution | 1000 |
| VGG19 | **Layer 30** | Convolution | 100352 |
| | **Layer 33** | Relu | 100352 |
| | **Layer 36** | Convolution | 100352 |
| | **Layer 38** | Max pooling | 25088 |
| AlexNet | **Layer 8** | Cross channel normalization | 186624 |
| | **Layer 9** | Max pooling | 43264 |
| | **Layer 10** | Convolution | 64896 |
| | **Layer 12** | Grouped convolution | 64896 |
| GoogleNet | **Layer 15** | Relu | 75264 |
| | **Layer 23** | Convolution | 25088 |
| | **Layer 40** | Max pooling | 94080 |
| | **Layer 70** | Relu | 25088 |
| MobileNetv2 | **Layer 58** | Clipped relu | 451584 |
| | **Layer 84** | Clipped relu | 75264 |
| | **Layer 150** | Clipped relu | 62720 |
| | **Layer 152** | Fully connected | 1000 |
| SqueezeNet | **Layer 20** | Convolution | 25088 |
| | **Layer 27** | Convolution | 25088 |
| | **Layer 32** | Relu | 100352 |
| | **Layer 34** | Max pooling | 50176 |
| DarkNet19 | **Layer 33** | Convolution | 65536 |
| | **Layer 39** | Convolution | 65536 |
| | **Layer 41** | Leaky relu | 65536 |
| | **Layer 45** | Max pooling | 32768 |
| ResNet50 | **Layer 44** | Convolution | 401408 |
| | **Layer 62** | Relu | 100352 |
| | **Layer 154** | Convolution | 25088 |
| | **Layer 175** | Fully connected | 1000 |
| ResNet101 | **Layer 68** | Addition | 401408 |
| | **Layer 91** | Relu | 401408 |
| | **Layer 142** | Convolution | 50176 |
| | **Layer 247** | Relu | 50176 |

The significance of bold is to emphasize the highest result

class [42]. The training of all N classes is accomplished by utilizing the rows of the ECOC matrix, where N denotes the dimensionality of new feature space and the length of codeword. If the trained ECOC SVM does not achieve satisfactory accuracy, the training parameters may be adjusted and the ECOC SVM is re-trained, as shown in Fig. 4.

- In testing, after being trained, the classifier is presented with the test data. The N classifiers identify the samples and produce an output vector Y. Equation (1) displays the Hamming distance between each vector in the output vector matrix and the code matrix. The classifier selects the class based on the smallest distance.

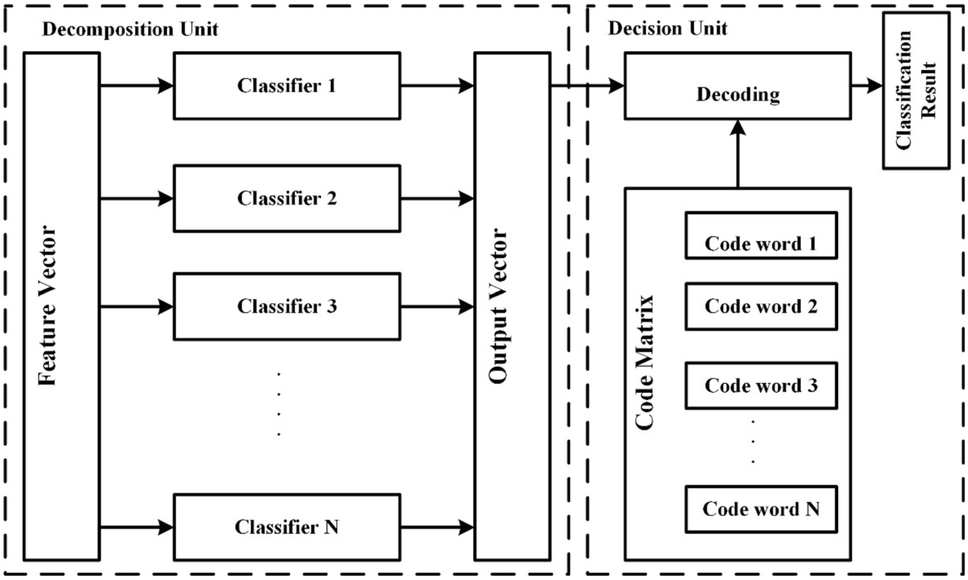


**Fig. 3** ECOC-SVM framework

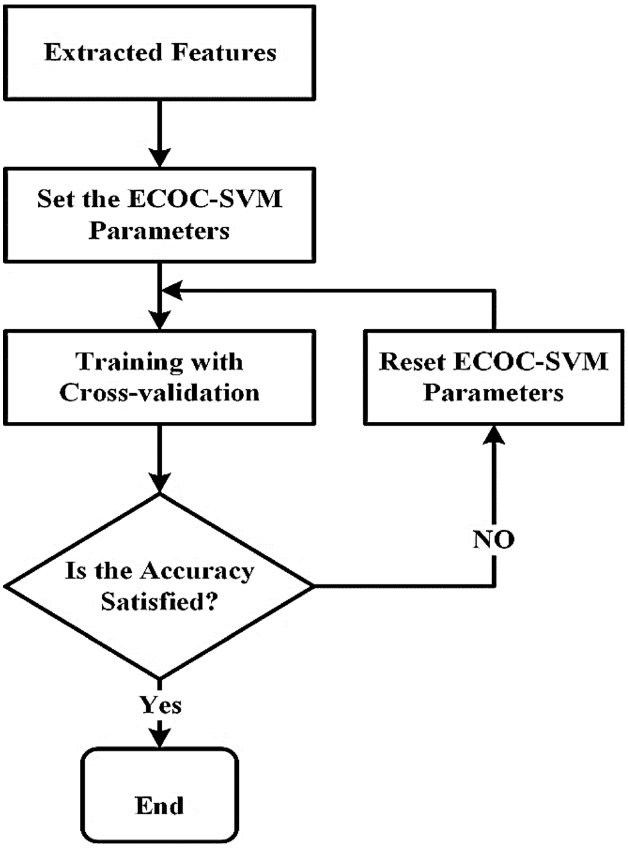

Extracted Features
Set the ECOC-SVM Parameters
Training with Cross-validation
Reset ECOC-SVM Parameters
Is the Accuracy Satisfied?
NO
Yes
End

**Fig. 4** ECOC-SVM Training process

$$\text{Argmind}(Y, H_i) = \sum_{j=1}^{N} \left| Y_j - H_{i,j} \right|, i = 1, 2, 3, \ldots.., m \tag{1}$$

*K*-nearest neighbor (KNN) is one of the easiest algorithms classified under the category of lazy algorithms. In KNN, the distance between two data points is typically measured using the Euclidean distance. Given a new data point, the KNN algorithm first identifies the k-nearest neighbors in the training set based on their distance to the new data point. The class label of the new data point is then assigned based on the most common class label among its *k*-nearest neighbors. In the case of a tie, the class with the closest neighbor is typically chosen [43]. Our work is based on a multiclass *k*-nearest neighbors (KNN) classifier which is a machine learning algorithm that can be used for classification tasks involving multiple classes. In this method, the classifier first determines the *k* nearest neighbors of a test instance from the training data based on a chosen distance metric. Then, the class label of the test instance is predicted based on the class labels of the *k* nearest neighbors using various techniques such as majority voting or weighted voting [44]. A *k* value which denotes the number of considered neighbors is calculated, and the Euclidean distance, which is the distance between two points as mentioned in Eq. (2) of the input from the *K*-nearest neighbors, is found.

$$\text{Euclidean distance} = \sqrt{\sum_{i=1}^{n} (Z_i - y_i)} \tag{2}$$

where $y_i$is the first point, and $Z_i$is the last point.

Then, the classifier can be represented by the following equation.

$$\vec{y_t}(l) = \arg \max_{b \int \{0,1\}} P\left(H_b^l \backslash E^{\overset{l}{\rightharpoonup}}_{c_t(l)}\right), l \in y \tag{3}$$

where $\vec{c}_t(l)$ is the membership counting vector, $\vec{y}_t(l)$ is the predicted category vector, and $P\left(H_b^l \backslash E^L_{c_t(l)}\right), l \in y$ is the prior probabilities for calculated neighbors.

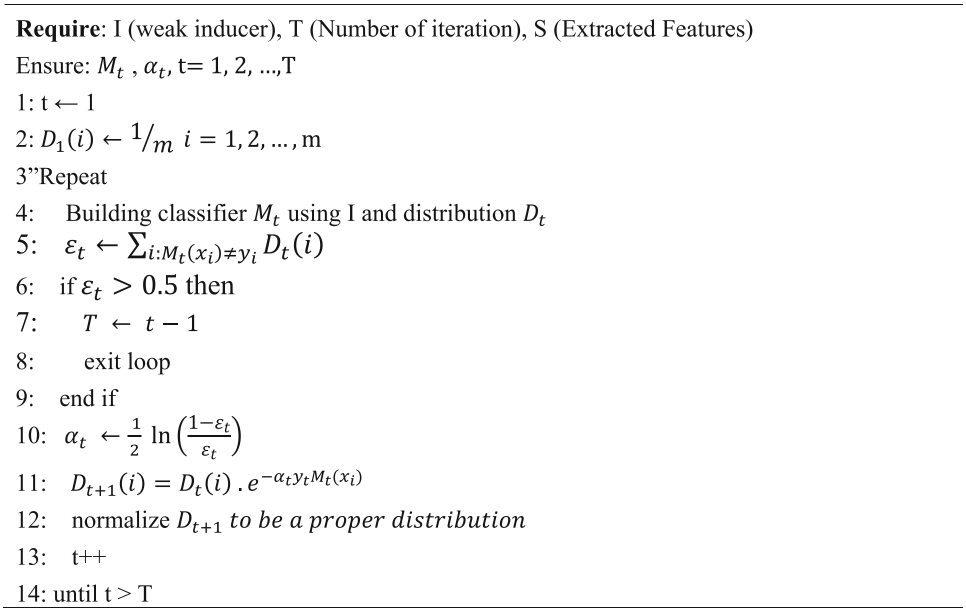
Ensemble Classifier is a supervised machine learning technique that combines the predictions of multiple base classifiers to improve the overall prediction accuracy. The idea behind an ensemble classifier is that the combination of several classifiers can result in a stronger and more accurate classifier than any individual classifier [45]. In our work, the algorithm is based on the principle of bagging, which involves constructing multiple versions of the training data by random sampling with replacement and training a base learner on each version [46]. The ensemble technique is based on the Adaboost method which improves the simple boosting algorithm through an iterative process and predicts new data using Eq. 4, as described in Algorithm 1. Two extension versions of the AdaBoost algorithm (AdaBoost.M1, AdaBoost.M2) enhance the ensemble classifier for binary and multiclass classification that is performed according to Eq. 5.

$$f(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t * M_t(x)\right) \tag{4}$$

$$f(x) = \underset{y \in dom(y)}{\arg\max}\left(\sum_{t:Mt(x)=y} \log \frac{1}{\beta_t}\right) \tag{5}$$

where $\beta_t \leftarrow \frac{1-\varepsilon_t}{\varepsilon_t}$.

**Algorithm 1** The Adaboost algorithm pseudo-code



```
Require: I (weak inducer), T (Number of iteration), S (Extracted Features)
Ensure: $M_t$ , $\alpha_t$, t= 1, 2, ...,T
t ← 1
$D_1(i) \leftarrow {}^{1}/_{m} \quad i = 1, 2, \ldots, m$
Repeat
    Building classifier $M_t$ using I and distribution $D_t$
    $\varepsilon_t \leftarrow \sum_{i:M_t(x_i) \neq y_i} D_t(i)$
    if $\varepsilon_t > 0.5$ then
        $T \leftarrow t - 1$
        exit loop
    end if
    $\alpha_t \leftarrow \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$
    $D_{t+1}(i) = D_t(i) \, . \, e^{-\alpha_t y_t M_t(x_i)}$
    normalize $D_{t+1}$ $to\ be\ a\ proper\ distribution$
    t++
until t > T
```

A Decision Tree (DT) is another ML classification algorithm that is a tree-like structure consisting of parent nodes, branches, and leaf nodes as shown in Fig. 5. The decision tree algorithm works by recursively splitting the data into subsets based on the values of the attributes, with the goal of maximizing the information gain or minimizing the impurity at each split [47]. The split criteria is the most critical aspect of a decision tree strategy. The internal nodes denote the attribute test, whereas branches denote the outcome [48]. The leaf nodes denote the class labels. At each node of the tree, the algorithm selects the feature that best separates the data into distinct classes. The algorithm then recursively splits the data into subsets based on the selected feature until it reaches a stopping criterion. This stopping criterion can be a maximum depth of the tree, a minimum number of samples per leaf, or other criteria [49]. A tree can be turned directly into a collection of rules in the context of decision tree learning, each of which is taken from a branch of the tree. The collection of regulations can be graded based on their quality. The confidence and J-measure are two prominent indicators of rule quality [50]. Confidence is a measure of a rule's weight (predictive accuracy), as described in Eq. 6.

$$\text{Confidence} = \frac{P(x, y)}{P(x)} \tag{6}$$

$P(x, y)$ is the joint probability that the antecedent and consequent of a rule both occur and $P(x)$ is the probability that the rule antecedent occurs.

The *J*-measure measures a rule's average information content that is the product of two terms described in Eq. 7. *J*-measure is also a prominent method of rating rules.

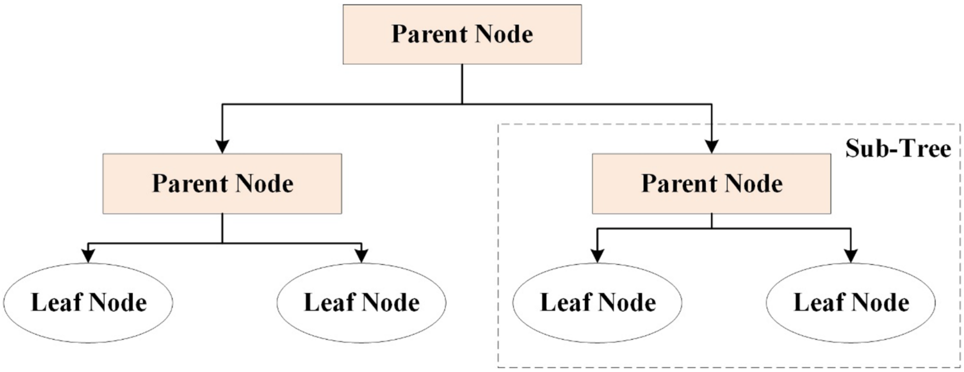


**Fig. 5** DT structure

$$J(Y, X = x) = P(x).j(Y, X = x) \tag{7}$$

The first term, taken as a measure of simplicity, is the chance that the antecedent (left hand side) of a rule happens. The second term is *j*-measure, which is a measure of a rule's goodness-of-fit, commonly known as cross entropy. Equation 8 defines the *j*-measure.

$$j(Y, X = x) = P(y|x).\log_2 \frac{P(y|x)}{p(x)} + (1 - P(y|x)).\log_2 \frac{1 - P(y|x)}{1 - p(x)} \tag{8}$$

NaiveBayes is a classification and regression ML algorithm that is based on supervised learning [51]. NaiveBayes classifier is a probabilistic classification algorithm that is based on Bayes' theorem and the assumption of independence between the features. The NaiveBayes classifier calculates the posterior probability of each class label given the feature values, and then assigns the label with the highest probability as the predicted class. Given a sample of dataset *X*, the classifier predicted that *X* belongs to the class *Ci* if

$$P(Ci|X) > P(Cj|X) \text{ for } 1 \leq j \leq m, j \neq i. \tag{9}$$

The equation for the NaiveBayes classifier can be written as in Eq. (10).

$$P(Ci|X) = \frac{P(X|Ci).P(Ci)}{P(X)} \tag{10}$$

Here $X$ is the data with an undefined class, $Ci$ is the hypothesis $X$ with a specific class, $P(Ci|X)$ is the probability of $Ci$ referring to $X$, $P(Ci)$ is the probability hypothesis $Ci$, $P(X|Ci)$ is the probability of $X$ in hypothesis $Ci$, and $P(X)$ is the probability of $X$.

## 4 Testing environment

In this section, we can spotlight the tools of our testing environment, such as the used machine, the input images dataset, and the performance metrics.

### 4.1 Machine

All the experiments in our work were carried out in MATLAB R2022a on an OMEN Laptop by HP with the following configuration: AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz, Nvidia Geforce RTX3070 8GB, and 16 GB RAM.

### 4.2 Dataset description

In our work, we used a dataset of 2482 CT scans downloaded from the Kaggle website [52], containing 1252 CT scans of positive COVID-19 and 1230 CT scans of non-COVID-19. These data were collected from real patients at hospitals in Sao Paulo, Brazil. We made our work based on three folds. This dataset is divided randomly between training and testing with respective percentage values (60%, 40%); (70%, 30%) and (80%, 20%), as shown in Table 2. The basic idea is to split the dataset into k equally sized folds. Then, the model is trained on the fold and validated on the remaining fold. This process is repeated *k* times, with each fold used once as a validation set. The advantage of *k*-fold cross-validation is that it allows for a more reliable estimate of a model's performance than a single holdout set. By averaging the performance across multiple folds, we obtain a more representative estimate of the model's expected performance on unseen data. As shown in Fig. 6, we can classify the input dataset into COVID and non-COVID classes.

### 4.3 Performance metrics

Some metrics can be used to evaluate the performance of our proposed model to classify COVID-19 by using CXR and CT datasets. These matrices are calculated based on the parameters of the confusion matrix: True Positive ($\alpha$), True Negative ($\beta$), False Positive ($\Omega$), and False Negative ($\mu$). These parameters are defined as follows:

- $\alpha$ denotes the number of COVID-19 images that are classified correctly as COVID-19.
- $\mu$ denotes the number of COVID-19 images that are incorrectly classified as non-COVID-19.
- $\Omega$ denotes the number of non-COVID-19 images that are incorrectly classified as COVID-19.
- $\beta$ denotes to the number of non-COVID-19 images that are classified correctly as non-COVID-19.

Accuracy (ACC) is the measure of correctly predicted cases related to the whole dataset. A high value of accuracy means more performance of the model [42].

$$\text{Accuracy(ACC)} = \frac{\alpha + \beta}{\alpha + \beta + \Omega + \mu} \tag{11}$$

Sensitivity (Sen) is considered to be the rate of correctly classified positive samples in the positive class dataset.

$$\text{Sensitivity (SEN)} = \frac{\alpha}{\alpha + \mu} \tag{12}$$

Specificity (SPC) is the measure of true negatives that are correctly classified by the model.

$$\text{Specificity (SPC)} = \frac{\beta}{\beta + \Omega} \tag{13}$$

Positive Predictive Value (PPV) is the proportion measure of positive classified cases that are truly positive.

$$\text{PPV} = \frac{\alpha}{\alpha + \Omega} \tag{14}$$

Negative Predictive Value (NPV) is the proportion measure of negative classified cases that are truly negative.

$$\text{NPV} = \frac{\beta}{\beta + \mu} \tag{15}$$

## 5 Results and discussion

In our study, we examined the performance of classification models for the detection of COVID-19 based on ten CNN models. In each model, we used five classifiers with randomly selected layers to check the best classifier with the best layer in each model. Then, we measured the performance of each classifier in terms of Accuracy, Sensitivity, Specificity, PPV, and NPV. In the case that models have the same accuracy value, we will check the performance based on higher sensitivity than specificity as it is better to classify non-patient to the patient than the patient to non-patient as it will help non-patients investigate more than cause the patient to die. The following results were performed using the dataset F-2.

In the Xception model, we will compare the performance results of the four layers that we selected. We excluded the results of NaiveBayes and Decision Tree classifiers as their results are negative. From Table 3, layer 79 achieved an accuracy value of 98.25% with the SVM classifier, 90.30% with the KNN classifier, and 91.73% with the ensemble classifier. Therefore, layer 79 has the highest accuracy value with the SVM classifier. Layer 108 achieved an accuracy value of 97.85% with the SVM classifier, 75.65% with the KNN classifier, and 91.90% with the ensemble classifier. Therefore, SVM could achieve the best accuracy value for layer 108. For layer 117, it could achieve an accuracy value of 98.25% with the SVM classifier, 78.33% with the KNN classifier, and 91.13% with the ensemble classifier. Therefore, layer 117 achieved the best value with SVM classifiers. Finally, for layer 134, an accuracy value of 97.85% is achieved with the SVM classifier, and 70.18% with the KNN classifier. However, for the ensemble classifier, there is no result for this layer due to more processing operations. The best performance for layer 134 is achieved with the SVM classifier. After this survey, we find that the best result for all selected layers is achieved with the SVM classifier. SVM could achieve the highest accuracy value with layer 79 and layer 117, which achieved the same values of accuracy. Therefore, to define the best layer, we can compare them by using other performance metrics. Layer 79 achieved a sensitivity value of 97.87%, whereas layer 117 achieved a sensitivity value of 98.13%. Therefore, we

**Table 2** The number of CT images used in each Fold

| Folds | Training | | Testing | | Total |
|---|---|---|---|---|---|
| | COVID-19 | Non-COVID-19 | COVID-19 | Non-COVID-19 | |
| Dataset F-1 | 752 | 736 | 500 | 494 | 2482 |
| Dataset F-2 | 876 | 860 | 376 | 370 | 2482 |
| Dataset F-3 | 1002 | 985 | 250 | 245 | 2482 |

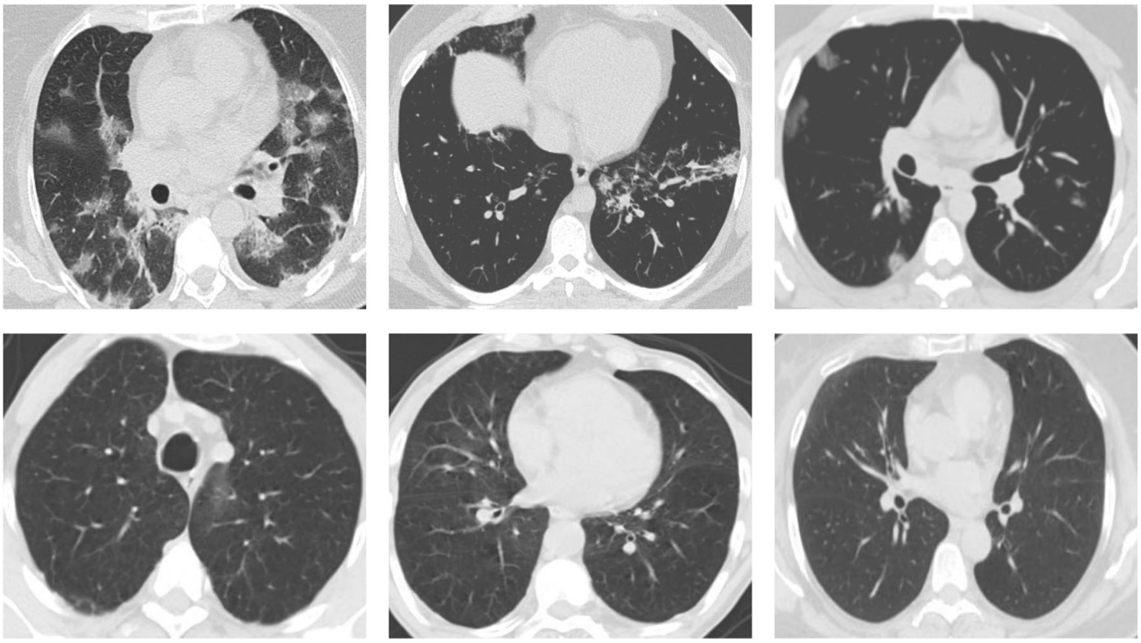


**Fig. 6** Different CT images are classified as COVID-19 in the above row and non-COVID-19 in the lower row

**Table 3** Performance metrics values of selected layers of the Xception Model with different classifiers

| Classifier | Metrics (%) | Layers | | | |
|---|---|---|---|---|---|
| | | **79** | **108** | **117** | **134** |
| SVM | **ACC** | 98.25 | 97.85 | **98.25** | 97.85 |
| | **SEN** | 97.87 | 97.60 | **98.13** | 97.66 |
| | **SPC** | 98.64 | 98.10 | **98.37** | 98.10 |
| | **PPV** | 98.65 | 98.12 | **98.40** | 98.17 |
| | **NPV** | 97.84 | 97.57 | **98.10** | 97.57 |
| KNN | **ACC** | 90.30 | 75.65 | 78.33 | 70.18 |
| | **SEN** | 86.90 | 52.97 | 60.59 | 42.14 |
| | **SPC** | 93.69 | 98.33 | 96.07 | 98.21 |
| | **PPV** | 93.23 | 96.94 | 93.91 | 95.93 |
| | **NPV** | 87.73 | 67.64 | 70.91 | 62.92 |
| Ensemble | **ACC** | 91.73 | 91.90 | 91.13 | – |
| | **SEN** | 89.76 | 87.61 | 88.21 | – |
| | **SPC** | 93.69 | 96.19 | 94.04 | – |
| | **PPV** | 93.43 | 95.83 | 93.67 | – |
| | **NPV** | 90.14 | 88.59 | 88.86 | – |

The significance of bold is to emphasize the highest result

can consider that the best model will be layer 117 with the SVM classifier to be selected as model 1.

For Darknet53, we will compare the results for selected four layers as shown in Table 4. We excluded NaiveBayes because of its bad results. Layer 32 achieved an accuracy value of 95.57% with the SVM classifier, 83.15% with the KNN classifier, 89.70% with the ensemble classifier, and no results for the decision tree classifier. Therefore, the best value is achieved by the SVM classifier. Layer 34 achieved an accuracy value of 95.43% with the SVM classifier, 72.98% with the KNN classifier, 89.76% with the ensemble classifier, and no result with the decision tree classifier. Therefore, the best value for layer 34 is achieved by the SVM classifier. Layer 42 achieved an accuracy value of 96.10%, 89.40% with the KNN classifier, 88.39% with the ensemble classifier, and 71.13% with the decision tree classifier. The best result for layer 42 is by the SVM classifier. For layer 182, an accuracy value of 92.21% is achieved by SVM, 89.29% with KNN, 91.13% with the ensemble, and 73.15% with the decision tree classifier. The best results for layer 182 are achieved by the SVM classifier. The SVM classifier could achieve the best performance results for all layers; however, the best values are for layer 42. Therefore, we can name the model of layer 42 with the SVM classifier as model 2.

For the VGG19 model, Table 5 shows the performance results for four layers. We exclude the results of NaiveBayes as their results are negative. For layer 30, 97.98% of the accuracy value is achieved by the SVM classifier, 86.79% with the KNN classifier, 95.18% with the ensemble classifier, and 74.88% with the decision tree classifier. Hence, the best results are achieved by the SVM classifier. Layer 33 achieved an accuracy value of 98.38% with SVM, 66.67% with KNN, 93.69% with the ensemble, and 75.12% with the decision tree. Therefore, layer 33 has the best performance values when using the SVM classifier. Layer 36 achieved an accuracy value of 97.04% with SVM, 90.95% with KNN, 92.44% with the ensemble, and 80.06%

**Table 4** Performance metrics values of selected layers of Darknet53 Model with different classifiers

| Classifier | Metrics (%) | Layers | | | |
|---|---|---|---|---|---|
| | | 32 | 34 | 42 | 182 |
| SVM | **ACC** | 95.57 | 95.43 | **96.10** | 92.21 |
| | **SEN** | 96.01 | 95.21 | **96.01** | 88.03 |
| | **SPC** | 95.48 | 95.66 | **96.20** | 96.47 |
| | **PPV** | 95.25 | 95.72 | **96.26** | 96.22 |
| | **NPV** | 96.21 | 95.14 | **95.94** | 88.77 |
| KNN | **ACC** | 83.15 | 72.98 | 89.40 | 89.29 |
| | **SEN** | 71.90 | 63.09 | 90.61 | 82.38 |
| | **SPC** | 94.40 | 82.85 | 89.16 | 96.19 |
| | **PPV** | 92.78 | 78.63 | 89.21 | 95.58 |
| | **NPV** | 77.06 | 69.18 | 90.56 | 84.51 |
| Ensemble | **ACC** | 89.70 | 89.76 | 88.39 | 91.13 |
| | **SEN** | 86.07 | 85.00 | 85.47 | 88.92 |
| | **SPC** | 93.33 | 94.52 | 91.30 | 93.33 |
| | **PPV** | 92.81 | 93.94 | 90.77 | 93.02 |
| | **NPV** | 87.01 | 86.30 | 86.27 | 89.39 |
| Decision tree | **ACC** | – | – | 71.13 | 73.15 |
| | **SEN** | – | – | 70.95 | 72.61 |
| | **SPC** | – | – | 71.30 | 73.69 |
| | **PPV** | – | – | 71.20 | 73.40 |
| | **NPV** | – | – | 71.05 | 72.90 |

The significance of bold is to emphasize the highest result

**Table 5** Performance metrics values of selected layers of the VGG19 Model with different classifiers

| Classifier | Metrics (%) | Layers | | | |
|---|---|---|---|---|---|
| | | 30 | 33 | 36 | 38 |
| SVM | **ACC** | 97.98 | **98.38** | 97.04 | 97.58 |
| | **SEN** | 97.87 | **98.93** | 97.34 | 97.87 |
| | **SPC** | 98.10 | **97.83** | 96.74 | 97.29 |
| | **PPV** | 98.13 | **97.89** | 96.82 | 97.35 |
| | **NPV** | 97.83 | **98.90** | 97.27 | 97.82 |
| KNN | **ACC** | 86.79 | 66.67 | 90.95 | 75.48 |
| | **SEN** | 78.69 | 34.28 | 87.73 | 59.16 |
| | **SPC** | 94.88 | 99.04 | 94.16 | 91.78 |
| | **PPV** | 93.89 | 97.29 | 93.76 | 87.80 |
| | **NPV** | 81.65 | 60.11 | 88.47 | 69.21 |
| Ensemble | **ACC** | 95.18 | 93.69 | 92.44 | 93.69 |
| | **SEN** | 94.16 | 92.61 | 90.35 | 92.85 |
| | **SPC** | 96.19 | 94.76 | 94.52 | 94.52 |
| | **PPV** | 96.11 | 94.64 | 94.28 | 94.43 |
| | **NPV** | 94.28 | 92.77 | 90.74 | 92.97 |
| Decision tree | **ACC** | 74.88 | 75.12 | 80.06 | 74.46 |
| | **SEN** | 73.57 | 74.16 | 80.71 | 75.23 |
| | **SPC** | 76.19 | 76.07 | 79.40 | 73.69 |
| | **PPV** | 75.55 | 75.60 | 79.67 | 74.09 |
| | **NPV** | 74.24 | 74.64 | 80.45 | 74.84 |

The significance of bold is to emphasize the highest result

with the decision tree classifier. Therefore, layer 36 has the best performance values when using the SVM classifier. Layer 38 could achieve an accuracy value of 97.58% with SVM, 75.48% with KNN, 93.69% with the ensemble, and 74.46% with the decision tree. With SVM, layer 38 could achieve its best performance. The SVM classifier could achieve the best performance for all selected layers; however, layer 33 could achieve the highest performance values with a 98.38% accuracy value. Therefore, we can name the model of layer 33 with SVM as model 3.

In AlexNet, the highest performance values are measured as shown in Table 6 for layers 8, 9, 10, and 12 and exclude the results of the NaiveBayes classifier as their results are negative. Layer 8 has an accuracy value of 96.37% with the SVM classifier, 77.20% with KNN, 92.02% with the ensemble, and 77.86% with the decision tree. Hence, the SVM classifier achieved the best performance for layer 8. Layer 9 achieved an accuracy of 97.44% with SVM, 83.33% with KNN, 92.38% with the ensemble, and 74.64% with the decision tree. Therefore, the best classifier for layer 9 is SVM. Layer 10 achieved an accuracy of 96.24% with SVM, 86.55% with KNN, 91.73% with the ensemble, and 76.61% with the decision tree. Therefore, the best classifier for layer 10 is SVM. Layer 12 achieved an accuracy of 97.04% % with SVM, 87.14% with KNN, 92.62% with the ensemble, and 76.37% with the decision tree. Therefore, the best classifier for layer 12 is SVM. SVM classifier could achieve a higher performance for all layers than other classifiers; however, layer 9 could achieve the highest performance between all layers with SVM. Therefore, we can consider layer 9 with the SVM classifier to be the best model, and name it model 4.

We can select the best layers in the case of GoogleNet with the highest performance values which are layers 15, 23, 40, and 70, as shown in Table 7. Layer 15 achieved 95.70% of accuracy with SVM, 79.40% with KNN, 90.12% with the ensemble, 71.19% with the decision tree, and no results for NaiveBayes. The best classifier for layer 15 is the SVM classifier. Layer 23 achieved an accuracy of 96.51% with SVM, 91.79% with KNN, 92.56% with the ensemble, 72.98% with the decision tree, and 85.36% with NaiveBayes. The best classifier for layer 23 is the SVM classifier. Layer 40 achieved an accuracy of 97.85% with SVM, 71.07% with KNN, 93.45% with the ensemble, and 76.07% with the decision tree, with no results for NaiveBayes. The best classifier for layer 23 is the SVM classifier. Layer 70 achieved an accuracy of 96.24% with SVM, 78.75% with KNN, 91.07% with the ensemble, 73.69%

**Table 6** Performance metrics values of selected layers of the Alexnet Model with different classifiers

| Classifier | Metrics (%) | Layers | | | |
|---|---|---|---|---|---|
| | | 8 | 9 | 10 | 12 |
| SVM | **ACC** | 96.37 | **97.44** | 96.24 | 97.04 |
| | **SEN** | 97.34 | **97.34** | 94.68 | 97.34 |
| | **SPC** | 95.39 | **97.56** | 97.83 | 96.74 |
| | **PPV** | 95.56 | **97.60** | 97.80 | 96.82 |
| | **NPV** | 97.23 | **97.29** | 94.75 | 97.27 |
| KNN | **ACC** | 77.20 | 83.33 | 86.55 | 87.14 |
| | **SEN** | 63.69 | 75.83 | 82.14 | 82.14 |
| | **SPC** | 90.71 | 90.83 | 90.95 | 92.14 |
| | **PPV** | 87.27 | 89.21 | 90.07 | 91.26 |
| | **NPV** | 71.41 | 78.98 | 83.58 | 83.76 |
| Ensemble | **ACC** | 92.02 | 92.38 | 91.73 | 92.62 |
| | **SEN** | 92.38 | 92.02 | 90.00 | 91.78 |
| | **SPC** | 91.66 | 92.73 | 93.45 | 93.45 |
| | **PPV** | 91.72 | 92.68 | 93.21 | 93.34 |
| | **NPV** | 92.32 | 92.08 | 90.33 | 91.92 |
| Decision tree | **ACC** | 77.86 | 74.64 | 76.61 | 76.37 |
| | **SEN** | 77.14 | 77.61 | 75.83 | 73.92 |
| | **SPC** | 78.57 | 71.66 | 77.38 | 78.80 |
| | **PPV** | 78.26 | 73.25 | 77.02 | 77.72 |
| | **NPV** | 77.46 | 76.20 | 76.20 | 75.14 |

The significance of bold is to emphasize the highest result

**Table 7** Performance metrics values of different selected layers of the GoogleNet Model with different classifiers

| Classifier | Metrics (%) | Layers | | | |
|---|---|---|---|---|---|
| | | 15 | 23 | 40 | 70 |
| SVM | **ACC** | 95.70 | 96.51 | **97.85** | 96.24 |
| | **SEN** | 97.34 | 96.27 | **97.87** | 95.47 |
| | **SPC** | 94.03 | 96.74 | **97.83** | 97.01 |
| | **PPV** | 94.32 | 96.79 | **97.87** | 97.02 |
| | **NPV** | 97.19 | 96.22 | **97.83** | 95.46 |
| KNN | **ACC** | 79.40 | 91.79 | 71.07 | 78.75 |
| | **SEN** | 64.28 | 91.78 | 45.35 | 64.28 |
| | **SPC** | 94.52 | 91.78 | 96.78 | 93.21 |
| | **PPV** | 92.15 | 91.78 | 93.38 | 90.45 |
| | **NPV** | 72.57 | 91.78 | 63.91 | 72.29 |
| Ensemble | **ACC** | 90.12 | 92.56 | 93.45 | 91.07 |
| | **SEN** | 89.88 | 90.83 | 93.09 | 89.52 |
| | **SPC** | 90.35 | 94.28 | 93.80 | 92.61 |
| | **PPV** | 90.31 | 94.08 | 93.76 | 92.38 |
| | **NPV** | 89.92 | 91.13 | 93.14 | 89.83 |
| De8cision tree | **ACC** | 71.19 | 72.98 | 76.07 | 73.69 |
| | **SEN** | 71.66 | 74.40 | 73.92 | 72.02 |
| | **SPC** | 70.71 | 71.54 | 78.21 | 75.35 |
| | **PPV** | 70.99 | 72.33 | 77.23 | 74.50 |
| | **NPV** | 71.39 | 73.65 | 75.00 | 72.92 |
| Naïve bayes | **ACC** | – | 85.36 | – | 50 |
| | **SEN** | – | 79.76 | – | 1 |
| | **SPC** | – | 90.95 | – | 0 |
| | **PPV** | – | 79.76 | – | 1 |
| | **NPV** | – | 81.79 | – | 1 |

The significance of bold is to emphasize the highest result

with the decision tree, and 50% for NaiveBayes. The best classifier for layer 23 is the SVM classifier. Therefore, between all used classifiers, SVM could achieve the best results for all layers, but layer 40 has the best values for SVM. Therefore, we can consider the model of layer 40 with the SVM classifier to be model 5 for the classification of COVID-19.

In the case of the Mobilenetv2 model, we concentrate on layers 58, 84, 150, and 152, as shown in Table 8. Based on five used classifiers, layer 58 has an accuracy value of 95.83% with SVM, 86.67% with KNN, 92.20% with the ensemble, 77.50% with Decision Tree, and 50% with NaiveBayes. Therefore, the best classifier for layer 58 is SVM. Layer 84 achieved a 98.79% accuracy value with SVM, 76.90% with KNN, 92.08% with the ensemble, 73.57% with Decision Tree, and 50% with NaiveBayes. Hence, SVM is the best classifier for layer 84. Layer 150 achieved an accuracy value of 96.64% with SVM, 70.71% with KNN, 90.18% with the ensemble, 71.55% with Decision Tree, and 50% with NaiveBayes. Therefore, the best classifier for layer 150 is SVM. Layer 152 achieved an accuracy value of 88.72% with SVM, 87.20% with KNN, 89.11% with the ensemble, 69.17% with Decision Tree, and 81.43% with NaiveBayes. Therefore, the best classifier for layer 150 is SVM. The best performance results for all layers are achieved with the SVM classifier and the worst values through NaiveBayes. For the SVM classifier, the layer that can achieve the highest values is layer 84. Therefore, we can consider layer 84 with SVM as our model 6 to classify COVID-19 disease.

In the Squeezenet model, the four layers that we selected to be tested with five ML classifiers are shown in Table 9. Layer 20 achieved an accuracy value of 96.10% with SVM, 86.49% with KNN, 90.95% with the ensemble, 75.18% with the decision tree, and 79.35% with NaiveBayes. Layer 20 could achieve the best performance with the SVM classifier. Layer 27 with the SVM classifier achieved 97.31% accuracy, 85.60% accuracy with KNN, 90.65% accuracy with the ensemble, 71.13% accuracy with the decision tree, and 80.48% with NaiveBayes. Therefore, Layer 27 could achieve the best performance result with the SVM classifier. Layer 32 achieved 96.51% of accuracy with SVM, 53.45% with KNN, 89.23% with the ensemble,

**Table 8** Performance metrics values of different selected layers of MobileNet Model with different classifiers

| Classifier | Metrics (%) | Layers | | | |
|---|---|---|---|---|---|
| | | 58 | 84 | 150 | 152 |
| SVM | **ACC** | 95.83 | **98.79** | 96.64 | 88.72 |
| | **SEN** | 97.34 | **99.20** | 97.07 | 82.44 |
| | **SPC** | 94.30 | **98.37** | 96.20 | 95.12 |
| | **PPV** | 94.57 | **98.41** | 96.30 | 94.51 |
| | **NPV** | 97.20 | **99.18** | 96.99 | 84.17 |
| KNN | **ACC** | 86.67 | 76.90 | 70.71 | 87.20 |
| | **SEN** | 78.57 | 58.57 | 46.90 | 84.04 |
| | **SPC** | 94.76 | 95.23 | 94.52 | 90.35 |
| | **PPV** | 93.75 | 92.48 | 89.54 | 89.70 |
| | **NPV** | 81.55 | 69.68 | 64.03 | 84.99 |
| Ensemble | **ACC** | 92.20 | 92.08 | 90.18 | 89.11 |
| | **SEN** | 90.11 | 89.76 | 88.57 | 88.80 |
| | **SPC** | 94.28 | 94.40 | 91.78 | 89.40 |
| | **PPV** | 94.03 | 94.13 | 91.51 | 89.34 |
| | **NPV** | 90.51 | 90.21 | 88.92 | 88.87 |
| Decision tree | **ACC** | 77.50 | 73.57 | 71.55 | 69.17 |
| | **SEN** | 77.50 | 72.73 | 75.83 | 70.11 |
| | **SPC** | 77.50 | 74.40 | 67.26 | 68.21 |
| | **PPV** | 77.50 | 73.97 | 69.84 | 68.80 |
| | **NPV** | 77.50 | 73.18 | 73.56 | 69.53 |
| Naïve bayes | **ACC** | 50 | 50 | 50 | 81.43 |
| | **SEN** | 1 | 1 | 1 | 72.97 |
| | **SPC** | 0 | 0 | 0 | 89.88 |
| | **PPV** | 1 | 1 | 1 | 72.97 |
| | **NPV** | – | – | – | 76.88 |

The significance of bold is to emphasize the highest result

**Table 9** Performance metrics values of different selected layers of the Squeezenet Model with different classifiers

| Classifier | Metrics (%) | Layers | | | |
|---|---|---|---|---|---|
| | | 20 | 27 | 32 | 34 |
| SVM | **ACC** | 96.10 | 97.31 | 96.51 | 97.98 |
| | **SEN** | 96.80 | 97.87 | 96.80 | 98.13 |
| | **SPC** | 95.39 | 96.74 | 96.20 | 97.83 |
| | **PPV** | 95.53 | 96.84 | 96.29 | 97.87 |
| | **NPV** | 96.70 | 97.80 | 96.73 | 98.09 |
| KNN | **ACC** | 86.49 | 85.60 | 53.45 | 85.36 |
| | **SEN** | 79.52 | 79.88 | 6.90 | 74.52 |
| | **SPC** | 93.45 | 91.30 | 1 | 96.19 |
| | **PPV** | 92.39 | 90.18 | 1 | 95.13 |
| | **NPV** | 82.02 | 81.94 | 51.78 | 79.06 |
| Ensemble | **ACC** | 90.95 | 90.65 | 89.23 | 93.39 |
| | **SEN** | 90.00 | 90.00 | 87.26 | 93.33 |
| | **SPC** | 91.90 | 91.30 | 91.19 | 93.45 |
| | **PPV** | 91.74 | 91.19 | 90.83 | 93.44 |
| | **NPV** | 90.18 | 90.12 | 87.74 | 93.34 |
| Decision tree | **ACC** | 75.18 | 71.13 | 68.51 | 74.17 |
| | **SEN** | 71.42 | 70.71 | 70.83 | 77.50 |
| | **SPC** | 78.92 | 71.54 | 66.19 | 70.83 |
| | **PPV** | 77.22 | 71.30 | 67.69 | 72.65 |
| | **NPV** | 73.42 | 70.95 | 69.41 | 75.89 |
| Naïve bayes | **ACC** | 79.35 | 80.48 | 50 | 50 |
| | **SEN** | 80.23 | 76.19 | 1 | 1 |
| | **SPC** | 78.45 | 84.76 | 0 | 0 |
| | **PPV** | 80.23 | 76.19 | 1 | 1 |
| | **NPV** | 79.87 | 78.07 | – | – |

The significance of bold is to emphasize the highest result

68.51% with the decision tree, and 50% with NaiveBayes. Layer 32 achieved higher performance results with the SVM classifier. Layer 34 achieved an accuracy value of 97.98% with SVM, 85.36% with KNN, 93.39% with the ensemble, 74.17% with the decision tree, and 50% with NaiveBayes. The best classifier for layer 34 is the SVM. We find that all layers achieved their best results with the SVM classifier. Therefore, for the SVM classifier, the layer with the best performance is layer 34 with an accuracy of 97.98%. Hence, layer 34 in Squeezenet Model with SVM has been selected to be model 7 due to achieving the highest performance.

With the Darknet19 model, layers achieving the highest performance results are mentioned in Table 10. For the SVM classifier, layer 33 achieved an accuracy value of 98.12%, layer 39 achieved a value of 97.85%, layer 41 achieved a 97.85% accuracy value, and layer 45 achieved a 98.25% accuracy value. Therefore, the best layer for SVM is layer 45. For the KNN classifier, layer 33 achieved an accuracy value of 89.52%, layer 39 achieved a value of 90.89%, layer 41 achieved an accuracy of 85.36%, and layer 45 achieved an accuracy of 81.73%. hence, layer 39 achieved the best result with the KNN classifier. For the ensemble classifier, layer 33 achieved 92.44% accuracy, layer 39 achieved 93.39% accuracy, layer 41 achieved 93.10% accuracy, and layer 45 achieved 93.33% accuracy. Therefore, the best layer for the ensemble classifier is layer 39. The decision tree classifier achieved an accuracy of 74.11% with layer 33, 72.08% with layer 39, 72.44% with layer 41, and 75.71% with layer 45. The decision tree classifier achieved the best performance result with layer 45. The NaiveBayes classifier achieved an accuracy of 77.74% with layer 33, 80.06% with layer 39, 50% with layer 41, and 50% with layer 45. The NaiveBayes classifier achieved the best performance with layer 39. For each layer, we can see that the highest result is achieved with the SVM classifier. Therefore, SVM is the best classifier for all layers, and layer 45 in Darknet 19 Model with SVM has

**Table 10** Performance metrics values of different selected layers of the Darknet19 Model with different classifiers

| Classifier | Metrics (%) | Layers | | | |
|---|---|---|---|---|---|
| | | 33 | 39 | 41 | 45 |
| SVM | **ACC** | 98.12 | 97.85 | 97.85 | **98.25** |
| | **SEN** | 98.67 | 97.07 | 97.87 | **97.87** |
| | **SPC** | 97.56 | 98.64 | 97.83 | **98.64** |
| | **PPV** | 97.63 | 98.64 | 97.87 | **98.65** |
| | **NPV** | 98.63 | 97.06 | 97.83 | **97.84** |
| KNN | **ACC** | 89.52 | 90.89 | 85.36 | 81.73 |
| | **SEN** | 86.30 | 92.14 | 79.64 | 73.09 |
| | **SPC** | 92.73 | 89.64 | 91.07 | 90.35 |
| | **PPV** | 92.23 | 89.89 | 89.91 | 88.34 |
| | **NPV** | 87.13 | 91.94 | 81.73 | 77.05 |
| Ensemble | **ACC** | 92.44 | 93.39 | 93.10 | 93.33 |
| | **SEN** | 91.30 | 93.69 | 92.26 | 91.42 |
| | **SPC** | 93.57 | 93.09 | 93.92 | 95.23 |
| | **PPV** | 93.42 | 93.13 | 93.82 | 95.04 |
| | **NPV** | 91.50 | 93.65 | 92.38 | 91.74 |
| Decision tree | **ACC** | 74.11 | 72.08 | 72.44 | 75.71 |
| | **SEN** | 71.90 | 69.64 | 70.00 | 77.38 |
| | **SPC** | 76.30 | 74.52 | 74.88 | 74.04 |
| | **PPV** | 75.21 | 73.21 | 73.59 | 74.88 |
| | **NPV** | 73.09 | 71.05 | 71.39 | 76.60 |
| Naïve bayes | **ACC** | 77.74 | 80.06 | 50 | 50 |
| | **SEN** | 83.21 | 77.50 | 1 | 1 |
| | **SPC** | 72.26 | 82.61 | 0 | 0 |
| | **PPV** | 83.21 | 77.50 | 1 | 1 |
| | **NPV** | 81.14 | 78.59 | – | – |

The significance of bold is to emphasize the highest result

been selected to be model 8 due to achieving the highest performance.

For the Resnet50 Model, the best results are achieved with layers 44, 62, 154, and 175 as shown in Table 11. Firstly, layer 44 achieved 94.36% accuracy with the SVM classifier, 87.68% with the KNN classifier, no result with the ensemble, no result with the decision tree, and 83.81% with NaiveBayes. The best classifier with layer 44 is SVM. Layer 62 had an accuracy of 95.57% with the SVM classifier, 70.71% with KNN, 85.59% with the ensemble, 70.24% with the decision tree, and 50% with NaiveBayes. Layer 62 achieved the best performance with the SVM classifier. Layer 154 achieved 98.25% accuracy with the SVM classifier, 97.84% with the KNN classifier, 94.23 with the ensemble, 75.36% with the decision tree, and 83.81% with NaiveBayes. The best classifier with layer 154 is SVM. Layer 175 has 90.46% accuracy with SVM, 88.45% with the KNN classifier, 89.76% with the ensemble, 73.51% with the decision tree, and 78.04% with NaiveBayes. Layer 175 achieved the best performance with the SVM classifier. Finally, we found that the SVM classifier could achieve the best result with all layers. Layer 154 in Resnet50 Model with SVM has been selected as model 9 as it achieved the highest performance.

The last model we tested was Resnet101 in which the best four layers are layers 68, 91, 142, and 247. Their performance results are mentioned in Table 12. Layer 68 achieved an accuracy of 95.83% with the SVM classifier, 82.92% with KNN, and no results with the ensemble, decision tree, and NaiveBayes. Layer 91 achieved an accuracy value of 97.98% with SVM, 85.54% with KNN, 92.50% with the ensemble, no result with the decision tree, and 50% with NaiveBayes. Layer 142 has an accuracy of 98.92% with SVM, 86.49% with KNN, 90.12% with ensemble classifier, 70.83% with the decision tree, and 82.74% with NaiveBayes. Layer 247 has an accuracy of 96.91% with SVM, 77.14% with KNN, 89.94% with ensemble classifier, 71.61% with decision tree classifier,

**Table 11** Performance metric values of different selected layers of the Resnet50 Model with different classifiers

| Classifier | Metrics (%) | Layers | | | |
|---|---|---|---|---|---|
| | | 44 | 62 | 154 | 175 |
| SVM | **ACC** | 94.36 | 95.57 | **98.25** | 90.46 |
| | **SEN** | 91.22 | 96.80 | **97.87** | 85.37 |
| | **SPC** | 97.56 | 94.30 | **98.64** | 95.66 |
| | **PPV** | 97.44 | 94.54 | **98.65** | 95.25 |
| | **NPV** | 91.60 | 96.66 | **97.84** | 86.51 |
| KNN | **ACC** | 87.68 | 70.71 | 92.02 | 88.45 |
| | **SEN** | 81.78 | 50.11 | 88.57 | 83.57 |
| | **SPC** | 93.57 | 91.30 | 95.47 | 93.33 |
| | **PPV** | 92.71 | 85.22 | 95.14 | 92.61 |
| | **NPV** | 83.70 | 64.67 | 89.30 | 85.03 |
| Ensemble | **ACC** | – | 89.70 | 94.23 | 89.76 |
| | **SEN** | – | 85.59 | 94.16 | 88.09 |
| | **SPC** | – | 93.80 | 94.28 | 91.42 |
| | **PPV** | – | 93.25 | 94.27 | 91.13 |
| | **NPV** | – | 86.68 | 94.17 | 88.47 |
| Decision tree | **ACC** | – | 70.24 | 75.36 | 73.51 |
| | **SEN** | – | 72.85 | 72.97 | 75.59 |
| | **SPC** | – | 67.61 | 77.73 | 71.42 |
| | **PPV** | – | 69.23 | 76.62 | 72.57 |
| | **NPV** | – | 71.35 | 74.20 | 74.53 |
| Naïve bayes | **ACC** | 83.81 | 50 | 83.87 | 78.04 |
| | **SEN** | 78.21 | 1 | 76.90 | 65.59 |
| | **SPC** | 89.40 | 0 | 90.83 | 90.47 |
| | **PPV** | 78.21 | 1 | 76.90 | 65.59 |
| | **NPV** | 80.40 | – | 79.72 | 72.45 |

The significance of bold is to emphasize the highest result and 50% with NaiveBayes. Layer 247 could achieve the best performance with SVM. For all tested layers, the SVM classifier achieved the best performance with layer 142 of the Resnet 101 model and has been selected to be model 10.

Now, we have 10 models that have the best performance results from all the tested models. These 10 models are tested with the three datasets F-1, F-2, and F-3, as shown in Table 13. We selected the best three models based on the highest accuracy as this is the most natural measure of performance. The best of the three models is model 10, which has the best accuracy with a value of 98.98%. This model uses layer 142 of Resnet101 to extract features. The next model is model 6 with an accuracy value of 99.39% which used layer 84 of the MobileNet model to extract features. The last model is model 3 with an accuracy value of 98.98%, which used layer 33 of the VGG19 model to extract features. All models used SVM as a classifier.

**Table 12** Performance metrics values of different selected layers of the Resnet101 Model with different classifiers

| Classifier | Metrics (%) | Layers | | | |
|---|---|---|---|---|---|
| | | 68 | 91 | 142 | 247 |
| SVM | **ACC** | 95.83 | 97.98 | **98.92** | 96.91 |
| | **SEN** | 96.27 | 97.34 | **99.20** | 98.40 |
| | **SPC** | 95.39 | 98.64 | **98.64** | 95.39 |
| | **PPV** | 95.51 | 98.65 | **98.67** | 95.60 |
| | **NPV** | 96.17 | 97.32 | **99.18** | 98.32 |
| KNN | **ACC** | 82.92 | 85.54 | 86.49 | 77.14 |
| | **SEN** | 70.47 | 74.52 | 81.42 | 61.30 |
| | **SPC** | 95.35 | 96.54 | 91.54 | 92.97 |
| | **PPV** | 93.81 | 95.57 | 90.59 | 89.72 |
| | **NPV** | 76.35 | 79.12 | 83.13 | 70.61 |
| Ensemble | **ACC** | – | 92.50 | 90.12 | 89.94 |
| | **SEN** | – | 89.88 | 88.09 | 90.23 |
| | **SPC** | – | 95.11 | 92.14 | 89.64 |
| | **PPV** | – | 94.84 | 91.81 | 89.70 |
| | **NPV** | – | 90.38 | 88.55 | 90.17 |
| Decision tree | **ACC** | – | – | 70.83 | 71.61 |
| | **SEN** | – | – | 73.92 | 75.35 |
| | **SPC** | – | – | 67.73 | 67.85 |
| | **PPV** | – | – | 69.61 | 70.09 |
| | **NPV** | – | – | 72.20 | 73.35 |
| Naïve bayes | **ACC** | – | 50 | 82.74 | 50 |
| | **SEN** | – | 1 | 81.42 | 1 |
| | **SPC** | – | 0 | 84.04 | 0 |
| | **PPV** | – | 1 | 81.42 | 1 |
| | **NPV** | – | – | 81.90 | – |

The significance of bold is to emphasize the highest result

To certify the credibility of the obtained results, a comparison with the results from the related work was performed in Table 14. We will compare these best 3 models with previous work related to the classification of COVID-19. The best performance results are achieved with model 10, model 6, and model 3. Model 3, Model 6, and Model 10 achieved an accuracy of 98.98%, 99.39%, and 98.98% with the dataset F-3, respectively. Model 10 has the highest average accuracy of 98.69%. However, Model 6 has the highest accuracy of 99.39% over all models. All the previous work mentioned achieved the highest accuracy of 98.66% in [30]. This is lower than our best three models as shown in Table 14.

### 5.1 Time consumption

Time consumption is a critical factor in many aspects of life. In the case of healthcare, time consumption can be a critical factor in determining patient outcomes, especially in emergencies. The time required for diagnostic tests and imaging can significantly impact the speed and accuracy of diagnosis and treatment. For example, in the case of COVID-19, delays in test results can lead to delayed treatment and increased spread of the disease. Moreover, imaging tests, such as CT scans, can take a significant amount of time to perform and interpret, which can delay diagnosis and treatment. However, technological advancements have led to the development of faster and more efficient diagnostic and imaging methods. Overall, time consumption is a critical factor in healthcare, and efforts to reduce it can lead to improved patient outcomes and overall healthcare efficiency. In our methodology, model 6, which achieved the best accuracy value among all models, could take a consumed time of 20.892 s in the training stage. Although the network training time was not considered, the average testing time per image was measured at 0.84 s.

## 6 Conclusion

Coronavirus disease (COVID-19), caused by SARS-CoV-2, is one of the major problems of the twenty-first century. COVID-19 has caused many injuries and deaths throughout the world over the previous 2 years. Computer-aided diagnosis has become an essential task in the fight against the virus's spread. Early detection of COVID-19 is critical for lowering patient mortality risk. Researchers are looking for quick answers using Machine Learning (ML) and Deep Learning approaches (DL). We provide a hybrid model for COVID-19 identification based on machine learning and deep learning models in this research. To extract characteristics from CT scans, we employed 10 distinct deep

**Table 13** the best 10 models with the highest performance using SVM

| Model NO | CNN Network | Layer | ACC (%) | | |
|---|---|---|---|---|---|
| | | | Dataset F-1 | Dataset F-2 | Dataset F-3 |
| Model 1 | Xception | 117 | 97.78 | 98.25 | 98.98 |
| Model 2 | DarkNet53 | 42 | 95.86 | 96.10 | 96.96 |
| **Model 3** | **VGG19** | **33** | **97.98** | **98.38** | **98.98** |
| Model 4 | AlexNet | 9 | 96.77 | 97.44 | 96.96 |
| Model 5 | GoogleNet | 40 | 97.68 | 97.85 | 98.98 |
| **Model 6** | **MobileNetV2** | **84** | **97.78** | **98.79** | **99.39** |
| Model 7 | SqueezeNet | 34 | 96.87 | 97.98 | 98.78 |
| Model 8 | DarkNet19 | 45 | **97.98** | 98.25 | 98.58 |
| Model 9 | ResNet50 | 154 | 97.37 | 98.25 | 98.98 |
| **Model 10** | **ResNet101** | **142** | **98.18** | **98.92** | **98.98** |

The significance of bold is to emphasize the highest result

**Table 14** Comparison between our models and old models

| Ref. No | ACC (%) | SEN (%) | SPC (%) | PPV (%) | NPV (%) |
|---|---|---|---|---|---|
| [23] | 91.53 | 91.68 | – | – | – |
| [24] | 90 | – | – | – | – |
| [25] | 95.33 | 95.33 | – | – | – |
| [27] | 93.73 | – | – | – | – |
| [27] | 94.74 | 91.0 | 98.89 | – | – |
| [28] | 95.1 | 95.1 | 95 | – | – |
| [29] | 98.05 | 90.99 | 95.39 | – | – |
| [30] | 98.66 | 97.76 | 100 | – | – |
| [31] | 98.25 | 95.30 | 97.60 | – | – |
| [32] | 97.68 | 96 | – | – | – |
| **Model 10** | **98.98** | **99.20** | **98.77** | **98.80** | **99.18** |
| **Model 6** | **99.39** | **99.60** | **99.18** | **99.20** | **99.59** |
| **Model 3** | **98.98** | **99.20** | **98.77** | **98.80** | **99.18** |

The significance of bold is to emphasize the highest result

CNN network models. We extract features from several levels in each network and determine the optimal layer that produces the best-extracted features for each CNN network. Then, to categorize these characteristics, we employed five different machine learning classifiers. The collection contains 2481 CT scans classified as COVID-19 and non-COVID-19, which is a public dataset from Kaggle. Three folds with various sizes are extracted between testing and training. Experiments are performed to determine the optimum layer for all CNN networks, the best network, and the best classifier. The measured performance demonstrated the suggested system's superiority over the literature, with the greatest accuracy of 99.39%. Our models have been evaluated with three folds, and the average accuracy between the three folds is 98.69%.

**Funding** Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

**Data availability** Data will be made available on reasonable request.

## Declarations

**Conflict of interest** The authors declare no competing interests.

## References

1. Alshazly H, Linse C, Abdalla M, Barth E, Martinetz T (2021) COVID-Nets: deep CNN architectures for detecting COVID-19 using chest CT scans. Peer J Comput sci. https://doi.org/10.7717/peerj-cs.655
2. Ji T, Liu Z, Wang G, Guo X, Akbar Khan S, Lai C, Chen H, Huang S, Xia S, Chen B, Jia H, Chen Y, Zhou Q (2020) Detection of COVID-19: A review of the current literature and future perspectives. Biosens Bioelectron 166:112455
3. Rong G, Zheng Y, Chen Y, Zhang Y, Zhu P, Sawan M (2021) COVID-19 diagnostic methods and detection techniques: a review. Ref Module Biomed Sci. https://doi.org/10.1016/B978-0-12-822548-6.00080-7

4. Skrobek D, Krzywanski J, Sosnowski M, Kułakowska A, Zylka A, Grabowska K, Ciesielska K, Nowak W (2020) Prediction of sorption processes using the deep learning methods (long short-term memory). Energies 13:6601
5. Krzywanski J, Blaszczuk A, Czakiert T, Rajczyk R, Nowak W (2014) Artificial intelligence treatment of NOx emissions from CFBC in air and oxy-fuel conditions. J Energy Eng 142(1):04015017
6. Muhammad Ashraf W, Moeen Uddin G, Hassan Kamal A, Haider Khan M, Khan AA, Afroze Ahmad H, Ahmed F, Hafeez N, Muhammad Zawar Sami R, Muhammad Arafat S, Gul Niazi S, Waqas Rafique M, Amjad A, Hussain J, Jamil H, Kathia MS, Krzywanski J (2020) Optimization of a 660 MWe supercritical power plant performance—a case of industry 4.0 in the data-driven operational management. part 2. power generation. Energies 13(21):5619
7. Janiesch C, Zschech P, Heinrich K (2021) Machine learning and deep learning. Electron Markets 31(3):685–695
8. Ghosh A, Sufian A, Sultana F, Chakrabarti A, De D (2020) Fundamental concepts of convolutional neural network. Recent trends adv artif intell Internet Things. https://doi.org/10.1007/978-3-030-32644-9_36
9. O'Shea V, Nash R (2015) An introduction to convolutional neural networks. ArXiv e-prints, 11/01
10. Kasban H, El-bendary M, Salama D (2015) A comparative study of medical imaging techniques. Int J Inf Sci Intell Syst 4:37–58
11. Ganguly D, Chakraborty S, Balitanas M, Kim T-H (2010) Medical imaging: a review. Springer, Berlin, pp 504–516
12. Miranda E, Aryuni M, Irwansyah E (2016) A Survey of Medical Image Classification Techniques. In: 2016 international conference on information management and technology (ICIMTech). IEEE
13. Yimer F, Tessema A, Simegn G (2021) Multiple lung diseases classification from chest x-ray images using deep learning approach. Int J Adv Trends Comput Sci Eng 10:2936–2946
14. Sadoon TA, Ali M (2020) An overview of medical images classification based on CNN. Int J Curr Eng Technol 10:900–905
15. Rushnaiwala T (2021) XCeption model
16. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767
17. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
18. Krizhevsky A (2014) One weird trick for parallelizing convolutional neural networks. arXiv preprint arXiv:1404.5997
19. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9
20. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L.-C (2018) Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4510–4520
21. Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K (2016) SqueezeNet: alexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size, arXiv preprint arXiv:1602.07360
22. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7263-7271
23. Al-Saffar B, Abdulmajeed NK COVID-19 Pandemic Detection in Chest X-ray Images by Deep Features with SVM Classifier
24. Hemdan EED, Shouman MA, Karar ME (2020) COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images. ArXiv, vol. abs/2003.11055
25. Sethy P, Santi K, Behera, Kumar P, Biswas P (2020) Detection of coronavirus Disease (COVID-19) based on Deep Features and Support Vector Machine, pp 643–651
26. Fernandez–Grandon C, Soto I, Zabala-Blanco D, Alavia W, Garcia V (2021) SVM and ANN classification using GLCM and HOG features for COVID-19 and Pneumonia detection from Chest X-rays. In: 2021 Third South American Colloquium on Visible Light Communications (SACVLC). IEEE, pp 01–06
27. Ismael AM, Şengür A (2021) Deep learning approaches for COVID-19 detection based on chest X-ray images. Expert Syst Appl 164:114054
28. Khan MA, Kadry S, Zhang Y-D, Akram T, Sharif M, Rehman A, Saba T (2021) Prediction of COVID-19 - pneumonia based on selected deep features and one class kernel extreme learning machine. Comput elect eng int j 90:106960
29. Aslan N, Dogan S, Ozmen Koca G (2022) Classification of chest X-ray COVID-19 images using the local binary pattern feature extraction method. Turk J Sci Technol 08(09):299–308
30. Hasoon JN, Fadel AH, Hameed RS, Mostafa SA, Khalaf BA, Mohammed MA, Nedoma J (2021) COVID-19 anomaly detection and classification method based on supervised machine learning of chest X-ray images. Res Phys 31:105045
31. Al-Azawi RJ, Al-Saidi NMG, Jalab HA, Kahtan H, Ibrahim RW (2021) Efficient classification of COVID-19 CT scans by using q-transform model for feature extraction. Peer J Comput Sci 7:e553
32. Kogilavani SV, Prabhu J, Sandhiya R, Kumar MS, Subramaniam U, Karthick A, Muhibbullah M, Imam SBS (2022) COVID-19 detection based on lung ct scan using deep learning techniques. Comput Math Methods Med 2022:7672196
33. Sharma S, Tiwari S (2021) COVID-19 Diagnosis using X-Ray Images and Deep learning. In: 2021 International conference on artificial intelligence and smart systems (ICAIS), pp. 344–349
34. Muhammad G, Shamim Hossain M (2021) COVID-19 and non-COVID-19 classification using multi-layers fusion from lung ultrasound images. Inf Fusion 72:80–88
35. Rahimzadeh M, Attar A (2020) A new modified deep convolutional neural network for detecting covid-19 from x-ray images
36. Yang S, Xiao W, Zhang M, Guo S, Zhao J, Shen F (2022) Image data augmentation for deep learning: a survey
37. Alsaffar A, Tao H, Talab M (2017) Review of deep convolution neural network in image classification. In: 2017 International conference on radar, antenna, microwave, electronics, and telecommunications (ICRAMET). IEEE
38. Tzotsos A, Argialas D (2008) Support vector machine classification for object-based image analysis. Springer, Berlin, pp 663–677
39. Khanna R, Awad M (2015) Efficient learning machines: theories, concepts, and applications for engineers and system designers. Springer, Berlin
40. Suthaharan S (2016) machine learning models and algorithms for big data classification: thinking with examples for effective learning. Integr Ser Inf Syst 36:1–2
41. Dietterich TG, Bakiri G (1994) Solving multiclass learning problems via error-correcting output codes. J artif intell res 2:263–286
42. Abd-Ellah M, Awad AI, Khalaf AAM, Hamed H (2018) Two-phase multi-model automatic brain tumor diagnosis system from magnetic resonance images using convolutional neural networks. EURASIP J Image Video Process 2018:97
43. Taunk K, De S, Verma S, Swetapadma A (2019) A Brief Review of Nearest Neighbor Algorithm for Learning and Classification. In: 2019 international conference on intelligent computing and control systems (ICCS). IEEE

44. Min-Ling Z, Zhi-Hua Z (2005) A k-nearest neighbor based algorithm for multi-label classification. In: 2005 IEEE international conference on granular computing, pp. 718–721 Vol. 2
45. Dietterich TG (2000) Ensemble methods in machine learning. Springer, Berlin, pp 1–15
46. Ponti M (2011) Combining Classifiers: From the Creation of Ensembles to the Decision Fusion, In: 2011 24th SIBGRAPI conference on graphics, patterns, and images tutorials. IEEE
47. Rokach L, Maimon O (2005) Decision trees. Springer, Berlin, pp 165–192
48. Jijo B, Mohsin Abdulazeez A (2021) Classification based on decision tree algorithm for machine learning. J Appl Sci Technol Trends 2:20–28
49. Colledani D, Anselmi P, Robusto E (2023) Machine learning-decision tree classifiers in psychiatric assessment: an application to the diagnosis of major depressive disorder. Psychiatry Res 322:115127
50. Liu H, Cocea M, Ding W (2017) Decision tree learning based feature evaluation and selection for image classification. In: 2017 International conference on machine learning and cybernetics (ICMLC) pp. 569–574
51. Wibawa A, Kurniawan A, Murti D, Adiperkasa RP, Putra S, Kurniawan S, Nugraha Y (2019) Naïve bayes classifier for journal quartile classification. Int J Recent Contrib Eng Sci IT (iJES) 7:91
52. Soares E, Angelov P, Biaso S, Froes MH, Abe DK (2020) SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification. In: medRxiv, pp 2020.04.24.20078584