



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

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



COVID-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X-ray images using fuzzy color and stacking approaches

Mesut Toğaçar^{a,*}, Burhan Ergen^b, Zafer Cömert^c

^a Department of Computer Technology, Vocational School of Technical Sciences, Firat University Elazig, Turkey

^b Department of Computer Engineering, Faculty of Engineering, Firat University Elazig, Turkey

^c Department of Software Engineering, Faculty of Engineering, Samsun University Samsun, Turkey

ARTICLE INFO

Keywords:

COVID-19

2019-nCoV

Fuzzy color technique

Stacking technique

Social mimic

Deep learning

ABSTRACT

Coronavirus causes a wide variety of respiratory infections and it is an RNA-type virus that can infect both humans and animal species. It often causes pneumonia in humans. Artificial intelligence models have been helpful for successful analyses in the biomedical field. In this study, Coronavirus was detected using a deep learning model, which is a sub-branch of artificial intelligence. Our dataset consists of three classes namely: coronavirus, pneumonia, and normal X-ray imagery. In this study, the data classes were restructured using the Fuzzy Color technique as a preprocessing step and the images that were structured with the original images were stacked. In the next step, the stacked dataset was trained with deep learning models (MobileNetV2, SqueezeNet) and the feature sets obtained by the models were processed using the Social Mimic optimization method. Thereafter, efficient features were combined and classified using Support Vector Machines (SVM). The overall classification rate obtained with the proposed approach was 99.27%. With the proposed approach in this study, it is evident that the model can efficiently contribute to the detection of COVID-19 disease.

1. Introduction

The new Coronavirus (COVID-19) is an acute deadly disease that originated from Wuhan province, China in December 2019 and spread globally. COVID-19 outbreak has been of great concern to the health community because no effective cure has been discovered [1]. The biological structure of COVID-19 comprises of a positive-oriented single-stranded RNA-type, and it is difficult to treat the disease owing to its mutating feature. Medical professionals globally are undergoing intensive research to develop an effective cure for the disease. Presently, COVID-19 is the primary cause of thousands of deaths globally, and major deaths are in the USA, Spain, Italy, China, the UK, Iran, etc. Many types of coronavirus exist, and these viruses are commonly seen in animals. COVID-19 has been discovered in human, bat, pig, cat, dog, rodent, and poultry. Symptoms of COVID-19 include sore throat, headache, fever, runny nose, and cough. The virus can provoke the death of people with weakened immune systems [2,3]. COVID-19 is transmitted from person to person mostly by physical contact. Generally, healthy people can be infected through breath contact, hand

contact, or mucous contact with people carrying COVID-19 [4].

Recently, artificial intelligence (AI) has been widely used for the acceleration of biomedical research. Using deep learning approaches, AI has been used in many applications such as image detection, data classification, image segmentation [5,6]. People infected by COVID-19 may suffer from pneumonia because the virus spreads to the lungs. Many deep learning studies have detected the disease using a chest X-ray image data approach [7]. A previous study has classified the pneumonia X-ray images using three different deep learning models [8] namely the fine-tuned model, model without fine-tuning, and the model trained from scratch. By using the ResNet model, they classified the dataset into multiple labels such as age, gender, etc. They also used the Multi-Layer Perceptron (MLP) as a classification method and achieved an average of 82.2% accuracy. Samir Yadav et al. [9] performed a classification algorithm using pneumonia data, SVM as a classification method, and InceptionV3, VGG-16 models as a deep learning approach. In their study, the dataset is divided into three classes: normal, bacterial pneumonia, and viral pneumonia to improve the contrast and brightness zoom settings with the augmentation method for each image in the

* Corresponding author.

E-mail addresses: mtogacar@firat.edu.tr (M. Toğaçar), bergen@firat.edu.tr (B. Ergen), zcomert@samsun.edu.tr (Z. Cömert).

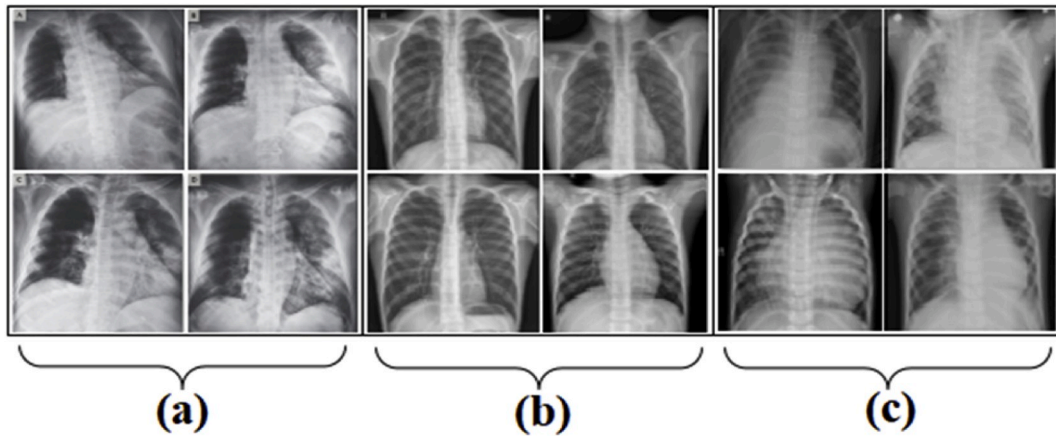


Fig. 1. The sample images used in the experimental analysis of this study; (a) COVID-19 chest images, (b) normal chest images, (c) pneumonia chest images.

dataset. The best classification achievement was 96.6%. Rahib Abiyev et al. [10] used the Backpropagation Neural Network and Competitive Neural Network models to classify pneumonia data. Using pneumonia and normal chest X-ray images, they set 30% of the dataset as test data and compared the proposed approach with the existing CNNs. They achieved 89.57% classification success. Okeke Stephen et al. [11] proposed a deep learning model to classify the pneumonia data from scratch to train the data. Their proposal consists of convolution layers, dense blocks, and flatten layers. The input size of the model was 200×200 pixels to determine the possibilities of classification using the sigmoid function. The success rate was 93.73% in pneumonia from X-ray images. Vikash Chouhan et al. [12] detected the images of pneumonia using deep learning models, three classes of the dataset: normal, virus pneumonia, and bacterial pneumonia images. In the first instance, they carried out a set of preprocessing procedures to remove noise from the images. Then, they applied the augmentation technique to each image and used a transfer learning to train the models. The overall classification accuracy was 96.39%.

In this study, we used COVID-19 chest images dataset, pneumonia chest images, and normal chest images. We preprocessed each image before being trained with deep learning models. In this preprocessing, the dataset was reconstructed using the Fuzzy technique and Stacking technique. Then, we trained the three datasets using the MobileNetV2 and SqueezeNet deep learning models and classified the models by the SVM method. The remainder of this study is structured as follows: In Section 2, we discuss the structure of the dataset, technique, method, deep learning models, and optimization algorithm. Experimental analysis is mentioned in Section 3. Section 4 and 5 present the discussion and conclusion, respectively.

2. Dataset, models, methods, and techniques

2.1. Dataset

In the experimental analysis, we use the three classes of datasets that are accessible publicly. These classes are normal, pneumonia, and COVID-19 chest images. All datasets are X-ray images, and each image is converted to JPG format. Since COVID-19 is a new disease, the number of images related to this virus is limited. For this study, we combined two publicly accessible databases consisting of COVID-19 images. The first COVID-19 dataset was shared on the GitHub website by a researcher named Joseph Paul Cohen from the University of Montreal. After the experts checked the images, they were made available to the public. In the Joseph Paul Cohen dataset, image types are MERS, SARS, COVID-19, etc. The data of 76 images labeled with COVID-19 were selected for this study [13]. The second COVID-19 dataset consists of the images created by a team of researchers from Qatar University, medical doctors from Bangladesh, and collaborators from Pakistan and Malaysia. The second COVID-19 dataset is available on the Kaggle website, and the current version has 219 X-ray images [14]. For this study, two datasets containing COVID-19 images were combined, and a new dataset consisting of 295 images was created. COVID-19 virus causes pneumonia in the affected individuals and can provoke death if the lungs are permanently damaged [15]. The second dataset is important in this study to compare COVID-19 chest images using deep learning models. The second dataset consists of normal chest images and pneumonia chest images. Pneumonia chest images include both virus and bacteria types, and these images are taken from 53 patients. The images were created by experts and shared publicly [16].

The combined dataset consists of three classes. Information about the classes of the dataset and the number of images in the classes are as follows: We collect a total of 295 images in COVID-19 class. Normal class X-ray images are 65 in total, and pneumonia class X-ray images are 98. The total number of images of the dataset is 458. In the experimental

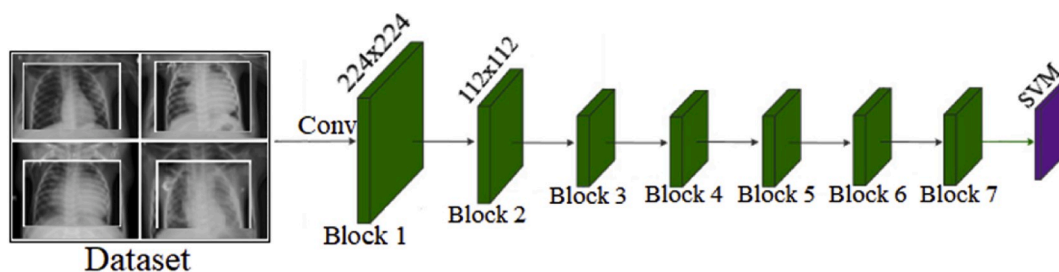


Fig. 2. The general design of the MobileNetV2 model used in this study [20].

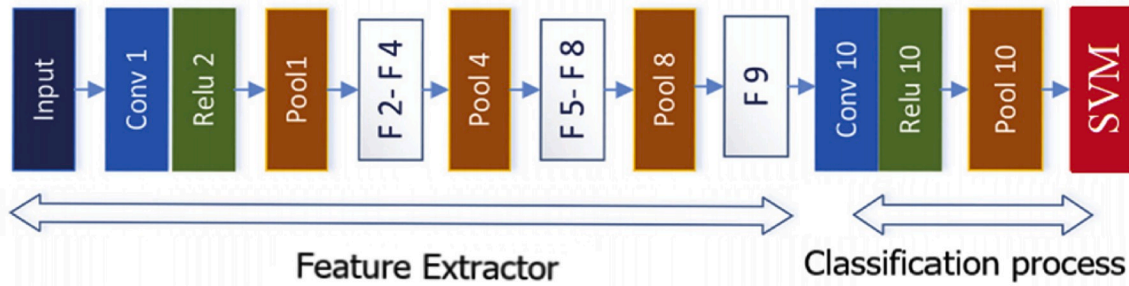


Fig. 3. The general design of the SqueezeNet model used in this study [21].

analysis, 70% of the dataset was used as training data, and 30% was used as test data. In the last step of the experiment, the k-fold cross-validation method was used for stacked images. Sample images of the dataset are shown in Fig. 1.

2.2. Deep learning model: MobileNetV2

MobileNet is a deep learning model intended to be used in low hardware cost devices. Object identification, segmentation, and classification can be performed using the MobileNet model. The MobileNet model is known as MobileNetV1, and the MobileNetV2 model is developed from the MobileNetV1. Compared to the MobileNetV2 model with the previous version, this new model offers the biggest contribution to the problems of linearity between layers. If linear bottlenecks occur between the layers, the problems are fixed in this version with shortcuts [17]. Fig. 2 shows the architecture design of the MobileNetV2 model. Its input size is 224 × 224 pixels, and its architecture comprises of in-depth

Table 1
General structure and parameters of MobileNetV2 architecture.

Type	Stride	Filter Size	Input Size
Convolution	2 × 2	3 × 3 × 3 × 32	224 × 224 × 3
Convolution DW	1 × 1	3 × 3 × 32	112 × 112 × 32
Convolution	1 × 1	1 × 1 × 32 × 64	112 × 112 × 32
Convolution DW	2 × 2	3 × 3 × 64	112 × 112 × 64
Convolution	1 × 1	1 × 1 × 64 × 128	56 × 56 × 64
Convolution DW	1 × 1	3 × 3 × 128	56 × 56 × 128
Convolution	1 × 1	1 × 1 × 128 × 128	56 × 56 × 128
Convolution DW	2 × 2	3 × 3 × 128	56 × 56 × 128
Convolution	1 × 1	1 × 1 × 128 × 256	28 × 28 × 128
Convolution DW	1 × 1	3 × 3 × 256	28 × 28 × 256
Convolution	1 × 1	1 × 1 × 256 × 256	28 × 28 × 256
Convolution DW	2 × 2	3 × 3 × 256	28 × 28 × 256
Convolution	1 × 1	1 × 1 × 256 × 512	14 × 14 × 256
5 × Convolution DW	1 × 1	3 × 3 × 512	14 × 14 × 512
5 × Convolution	1 × 1	1 × 1 × 512 × 512	14 × 14 × 512

(DW) separable filters and combination of steps. The model performance increases as it examines the DW, and the input features are divided into two layers. Each layer is subdivided into the next layer by combining it with the output features until the process is completed. MobileNetV2 model uses the ReLU function between layers [18]. Thus, it enables the nonlinear outputs from the previous layer to be linearized and transferred as input to the next layer. The model continues its training process until a comfortable step is reached. In this model, the convolutional layers circulate filters over input images and create activation maps. The

Table 2
Other parameters of the MobileNetV2 model and preferred values in this study.

Software Used	Model	Image Size	Optimization	Momentum	Decay	Beta	Mini Batch	Learning Rate
MATLAB	MobileNetV2	224 × 224	Stochastic Gradient Descent (SGD)	0.9	1e-6	–	64	10 ⁻⁵

activation maps contain the features extracted from the input images, and these features are transferred to the next layer. Pooling layers are also used in the MobileNetV2 model. The matrices obtained through these layers are converted into smaller dimensions [19].

MobileNetV2 model was used as pre-trained, and the SVM method was used in the classification phase. Besides, other important parameters of the MobileNetV2 model are given in Table 1 and Table 2. All default parameter values were used for the MobileNetV2 model without any change.

2.3. Deep learning model: SqueezeNet

SqueezeNet, an in-depth learning model of input size of 224 × 224 pixels, comprises of convolutional layers, pooling layers, ReLU, and Fire layers. The SqueezeNet does not have fully connected layers and dense layers. However, Fire layers perform the functions of these similar layers. The major benefit of this model is that it performs analyses successfully by reducing the number of parameters, thereby decreasing the model size capacity. SqueezeNet model produced more successful results, approximately 50 times fewer parameters than the AlexNet model, thereby reducing the cost of the model [21]. Fig. 3 presents the model design.

While the information about the layers is presented in the MobileNetV2 model, the Fire (F2, F3, ..., F9) layers that look like a new layer consisting of two parts namely the Compression and Expansion parts. This model uses only a 1 × 1 convolutional filter to the input image in the Compression part of the Fire layer. In the Expansion part, it uses both 1 × 1 and 3 × 3 convolutional filters to the input image. The Compression part and the Expansion part keep the same feature map size. In the Compression part, the depth of the input image is reduced and then increased (bottleneck). In the Expansion part, the depth is increased [21,22]. Table 3 presents the layers and default parameter values of the model, these values are used without changes. Other important parameters of the SqueezeNet model are given in Table 4.

2.4. Classification method: SVM and optimization method: SGD

SVM is a machine learning used for regression and classification analysis. This method uses a hyper-plane line in the classification process to separate the features in the data classes. It chooses a location away from the features of the classes to determine a line. The distances of each class are measured according to the hyper-plane determined by the SVM method. The one with the highest voting score is transferred to the class labeled as the highest voting score. Fig. 4 shows the design of the SVM method for the classification process. The functions of the

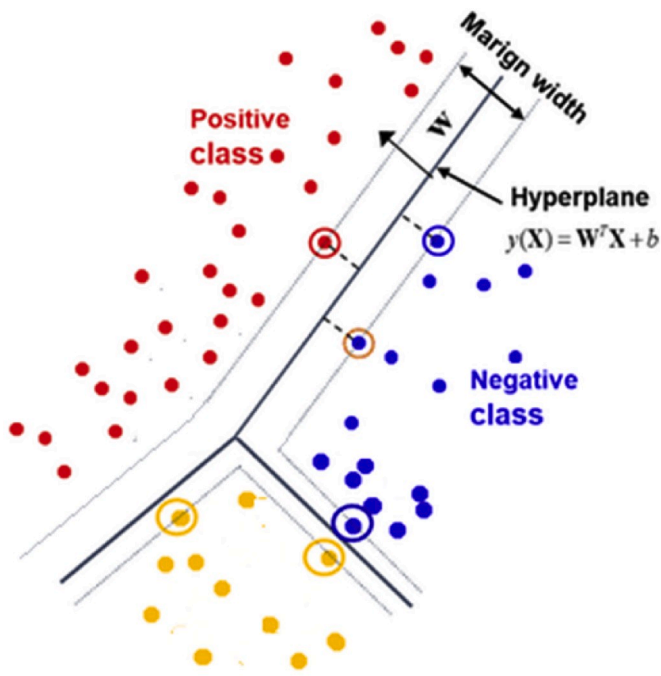


Fig. 4. Design of SVM method for multiple classification process.

$$u = \vec{w} \cdot \vec{x} - b \tag{1}$$

$$\frac{1}{2} \|\vec{w}\|^2 \tag{2}$$

$$y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1, \forall i \tag{3}$$

Table 3
General structure and parameters of SqueezeNet architecture.

Type	Stride	Filter Size	Output Size
Input	–	–	224 × 224 × 3
Convolution	2	96 × 96 × 7	109 × 109 × 96
Pooling	2	3 × 3	54 × 54 × 96
Fire 2	–	16 × 16 × 1, 64 × 1 × 1, 64 × 3 × 3	54 × 54 × 128
Fire 3	–	16 × 16 × 1, 64 × 1 × 1, 64 × 3 × 3	54 × 54 × 128
Fire 4	–	32 × 1 × 1, 128 × 1 × 1, 128 × 3 × 3	54 × 54 × 256
Pooling	2	3 × 3	27 × 27 × 256
Fire 5	–	32 × 1 × 1, 128 × 1 × 1, 128 × 3 × 3	27 × 27 × 256
Fire 6	–	48 × 1 × 1, 192 × 1 × 1, 192 × 3 × 3	27 × 27 × 384
Fire 7	–	48 × 1 × 1, 192 × 1 × 1, 192 × 3 × 3	27 × 27 × 384
Fire 8	–	64 × 1 × 1, 256 × 1 × 1, 256 × 3 × 3	27 × 27 × 512
Pooling	2	3 × 3	13 × 13 × 128
Fire 9	–	64 × 1 × 1, 256 × 1 × 1, 256 × 3 × 3	13 × 13 × 512
Convolution	1	6 × 13 × 13	13 × 13 × 6
Pooling	–	13 × 13	1 × 1 × 6

mathematical operations are given in Eq. (1), Eq. (2) and Eq. (3). Here, X and Y represent the coordinate points of the features in the hyperplane. W parameter represents margin width, and b parameter represents bias value [23,24].

Deep learning models use the optimization methods, and these facilitate learning trends of models. Stochastic Gradient Descent (SGD) optimization is the method that updates the weight parameters in the

Table 4
Other parameters of the SqueezeNet model and preferred values in this study.

Software Used	Model	Image Size	Optimization	Momentum	Decay	Beta	Mini Batch	Learning Rate
MATLAB	SqueezeNet	224 × 224	SGD	0.9	1e-6	–	64	10 ⁻⁵

model structure at every iteration. The models provide better training during each iteration. However, the SGD does not use all the images input into the model while updating the parameters. It performs this operation using only the images it randomly determines. This lowers the cost of the model, and it offers a faster training process. Eq. (4) shows the mathematical function that performs the weight parameter updates in the SGD. Where Θ represents the weight parameter. This represents the coordinates of the features extracted in the X and Y input images and represents α learning rate [25].

$$\Theta_i = \Theta_{i-1} - \alpha \nabla_{\Theta} J(\Theta; x^i, y^i) \tag{4}$$

In this study, the SVM method was preferred because:

- i. It has a strong potential to provide solutions to the data analysis problems encountered in daily life,
- ii. It is widely used for the remote pattern recognition and classification problems to successfully execute multiple classification processes [26,27], and
- ii. It gives the best classification performance among other machine learning methods (discriminant analysis, nearest neighbor, etc.).

Moreover, the Linear SVM was preferred owing to its best performance such as cubic, linear, quadratic, etc. Preferred parameter values in the Linear SVM method were the kernel scale that was parameter automatically selected. The box constraint level parameter value was chosen, and the multiclass method parameter of one-vs-one was selected.

2.5. Reconstructing images: fuzzy color technique

The Fuzzy concept is accepted based on its degree of accuracy, and its next degree is uncertain. Fuzzy Color algorithms play an important role in image analysis, and the obtained results depend on the similarity/difference functions used for color separation. In the fuzzy color technique, each of the input images contains three input variables (red, green, and blue - RGB). As a result of this process, a single output variable is passed. The input and output values are determined according to the training data [28,29].

The logic behind the Fuzzy Color technique is to separate the input data into blurred windows. Each pixel in the image has a membership degree to each window, and membership degrees are calculated based on the distance between the window and the pixel. Image variance is obtained with membership degrees. The Fuzzy Color technique is to create the finishing output-input. In this step, the weights of the images of each blurred window are summed, and the output image is created from the average. Here, the weight value of each pixel is expressed as the degree of membership [28,30]. We recreated the original dataset using the Python codes with the Fuzzy Color technique [31]. Fig. 5 shows the structure of the data image.

2.6. Reconstructing images: stacking technique

Image stacking is a digital image processing technique that combines multiple images shot or is reconstructed at different focal distances. This is a technique used to improve the quality of the images in the dataset. This technique aims to eliminate the noises from the original image by combining at least two images in a row and dividing the image into two parts. These parts are background and overlay. While the first images are processed in the background, the second is overlaid on the image placed

in the background. Here, parameters such as opacity, contrast, brightness, and combining ratio of the two images are important. The more accurately these ratios, the more amount of noise is reduced in the images, and the higher the quality ratio [32].

In this study, Python and Pillow library was used for the Stacking Technique [33]. Here, the original dataset was stacked on the reconstructed dataset using the Fuzzy technique. The successful result of the Fuzzy technique will contribute to the success of the stacking technique. The parameter values preferred in the stacking technique were the opacity value, with the value of 0.6, the contrast value was 1.5, the brightness value was set to -80 , and the combined ratio was chosen as 50%. These values can be varied for another dataset. We evaluate the various stages for dataset images and determine that these values are the most efficient of the dataset. Hence, we used them in the experimental analysis. Moreover, the original dataset was placed in the background, and the structured dataset was placed in the overlay. A combined representation of the original dataset using the stacking technique and dataset structured with the Fuzzy technique is shown in Fig. 6.

2.7. Social Mimic Optimization

Emotions such as morale and happiness can be transfer from person to person. Emotional imitation happens consciously or unconsciously. This situation is related to neurons in the human brain, that is, the nervous system. Some of these neurons are responsible for perceiving facial expressions such as our eyebrows sharpening or laughing, etc. These neurons also act as interbrain communication and generate a response to this effect. In other words, social imitation is the condition of adopting the behavior, conversation, or dressing of another person, which one sees as a guide. Based on this, an advanced is made towards a better understanding of human behavioral activities [34].

Social Mimic Optimization (SMO) is a method developed by inspiring people to imitate other people. This is inspired by imitating the behavior of people in society. Locally, each problem produces a solution using the SMO algorithm, which moves towards the global solution. Each solution determines the difference in the global value by comparing the local value obtained in the last iteration. This is applied to the problem parameters to find the solution randomly based on the difference in the value obtained. While the term "Follower" in SMO expresses the population, the "Leader" parameter expresses the best global value, and the "number of imitations" parameter represents the number of iterations. The number of followers is obtained by multiplying the population with the number of decision variables (N). Decision variables have a lower bound (lb) and upper bound (ub) value. Eqs. (5)–(7) are used to implement the SMO algorithm [34]. For this study, the population size of 20 was selected, the maximum iteration parameter was selected as 10. The best global value parameter was taken as 1000 for the start, and the N parameter value of 10 was chosen.

$$Follower_i(1,j) = lb_j + Rand \times (ub_j - lb_j); j = 1, 2, \dots, N \quad (5)$$



Fig. 5. Sub-data samples of the original dataset obtained by the Fuzzy Color technique.

$$Difference = \frac{(Leader - Fitness\ of\ (Follower_i))}{Fitness\ of\ (Follower_i)} \quad (6)$$

$$Follower_i = Follower_i + Difference \times Follower_i \quad (7)$$

2.8. Proposed approach

The proposed approach is designed to perform the classification of chest images based on the dataset types. This is an approach to separate chest images of COVID-19 virus infection from normal breast images and pneumonia. The original dataset is passed through the front image processing steps. In the first step, the original dataset was recreated with the Fuzzy Color technique, aimed to remove the noise in the original images. In the second step, using the original dataset Stacking technique, each fuzzy color image was combined with original images, and a new dataset was created. The aim was to create a better data quality image. The two deep learning models were used, and the stacked dataset was trained with MobileNetV2 and SqueezeNet deep learning models. Using the SMO algorithm, the 1000-features obtained by the models were used to extract efficient features. By combining the efficient features, the SVM method that produced successful results in multiple classifications was used as the classification process. Fig. 7 shows the overall design of the proposed approach.

3. Experimental analysis and results

Python 3.6 is used to structure the original data set using the Fuzzy technique and Stacking technique. Besides, the SMO algorithm was compiled in Python, and detailed information about the source codes and analysis used in this study are given in the web link specified in the Open Source Code section. Jupyter Notebook is the interfaces program used in compiling Python. Using deep learning models, MATLAB (2019b) software was used for classification. The hardware features to compile software are the Windows 10 operating system (64 bit) with a 1 GB graphics card, 4 GB memory card, and an Intel © i5 - Core 2.5 GHz processor.

The performance metrics derived from the confusion matrix are used for the experimental analysis. Eq. (8) to Eq. (12) were used to calculate these metrics. These metrics are Sensitivity (Se), Specificity (Sp), F-score (F-Scr), Precision (Pre), and Accuracy (Acc). True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) parameters of the confusion matrix were used to calculate the metrics [35,36]. When the samples that belong to a specific class are identified correctly by the classifier, these samples are located to TP indices. The other samples that belong to the other classes identified correctly are in the TN indices in the confusion matrix. Similarly, FP and FN indices in the confusion matrix correspond to the numbers of the samples incorrectly predicted by the classifier.

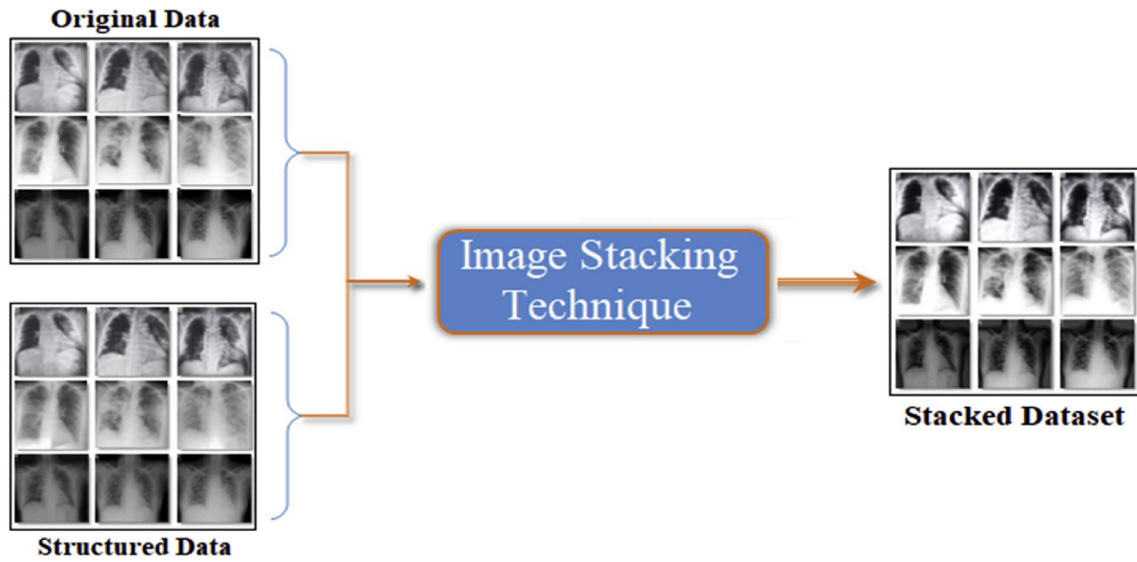


Fig. 6. Sub-image samples obtained by the Stacking technique.

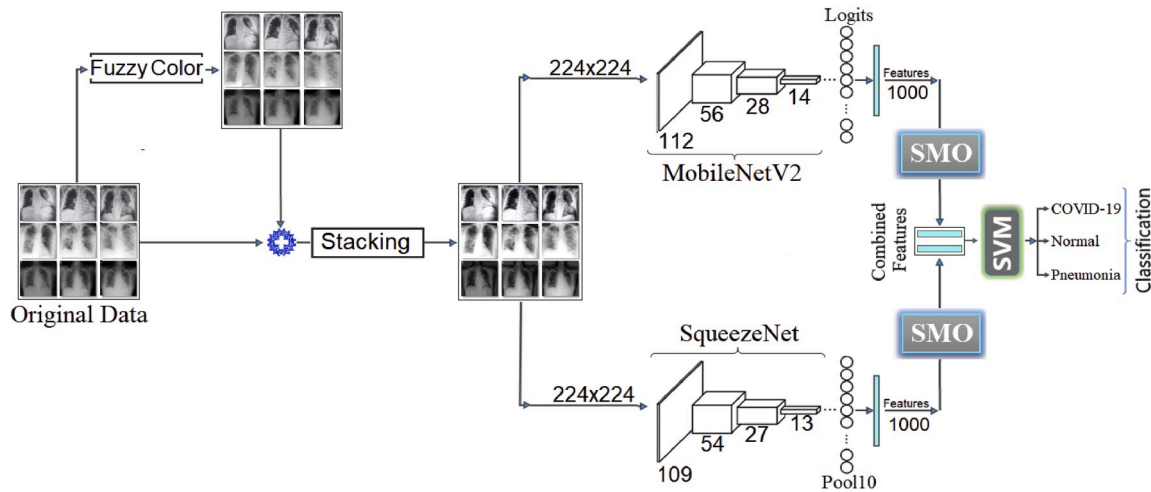


Fig. 7. The general design of the proposed approach.

$$Se = \frac{TP}{TP + FN} \tag{8}$$

$$Sp = \frac{TN}{TN + FP} \tag{9}$$

$$Pre = \frac{TP}{TP + FP} \tag{10}$$

$$F - Scr = \frac{2xTP}{2xTP + FP + FN} \tag{11}$$

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

The experiment consists of three steps. The 30% of the data set was used as test data and the remaining 70% as training data. In the steps related to the stacked dataset, the k-fold cross-validation method was used. The SVM method was used as a classifier in the last layers of SqueezeNet and MobileNetV2 models.

In each model, the first step consists of three stages, and each CNN model is trained with the original dataset, the dataset structured by the Fuzzy Color technique, and the stacked dataset classified by the SVM method. The overall accuracy rate of the SqueezeNet model in the

classification of the original dataset was 84.56%. In the second stage performed with the SqueezeNet model, the dataset structured using the Fuzzy Color technique was classified with a 95.58% overall accuracy rate. In comparison, the two stages of the Fuzzy Color technique contributed to the training of the SqueezeNet model. In the third stage, realized with the SqueezeNet model, the stacked dataset was trained with the 97.06% classification success (overall accuracy). In the training of this model, the structured and stacked datasets are used in achieving this success. Fig. 8 shows the training success graphs of the three stages performed with the SqueezeNet model, and Fig. 9 shows the confusion matrices. The results of the experimental analysis are given in Table 5. To acknowledge the validity of these analyses, we conducted a new analysis using another deep learning model, MobileNetV2. In the first stage of the MobileNetV2 model, we trained the original dataset, and the overall accuracy rate obtained with the SVM method was 96.32%. In the second stage, the dataset structured with the Fuzzy Color technique was classified and the overall accuracy rate obtained in this classification was 97.05%. In the third stage of the MobileNetV2 model, the model was trained with the stacked data set, and the overall accuracy rate in the classification was 97.06%. Fig. 10 shows the graphs of the training success of the three stages performed with the MobileNetV2 model, and Fig. 11 shows the confusion matrices. The results of the experimental

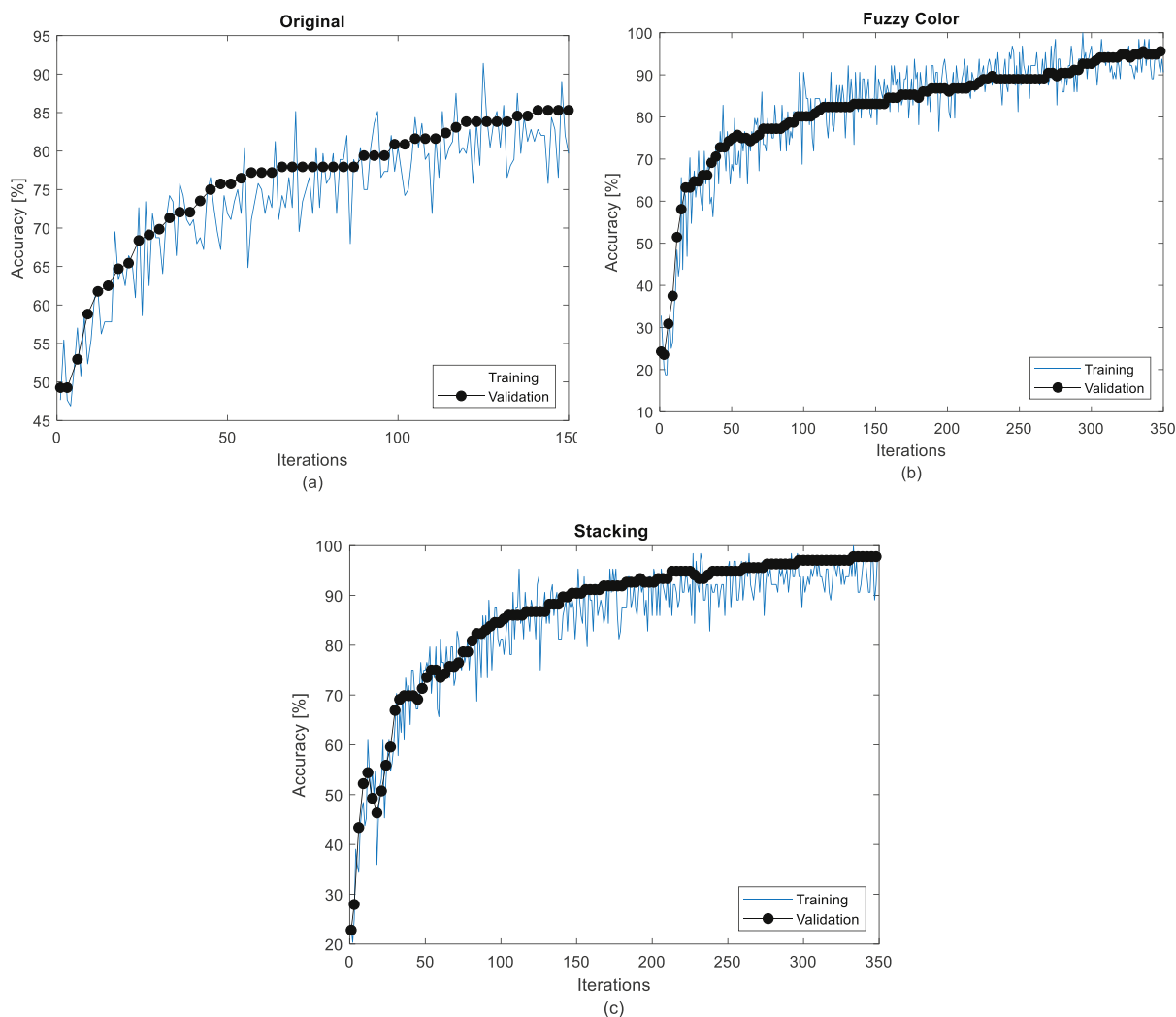


Fig. 8. Training and test success graphs of the SqueezeNet model; (a) original dataset, (b) dataset restructured using the Fuzzy technique, (c) dataset combined using the Stacking technique.

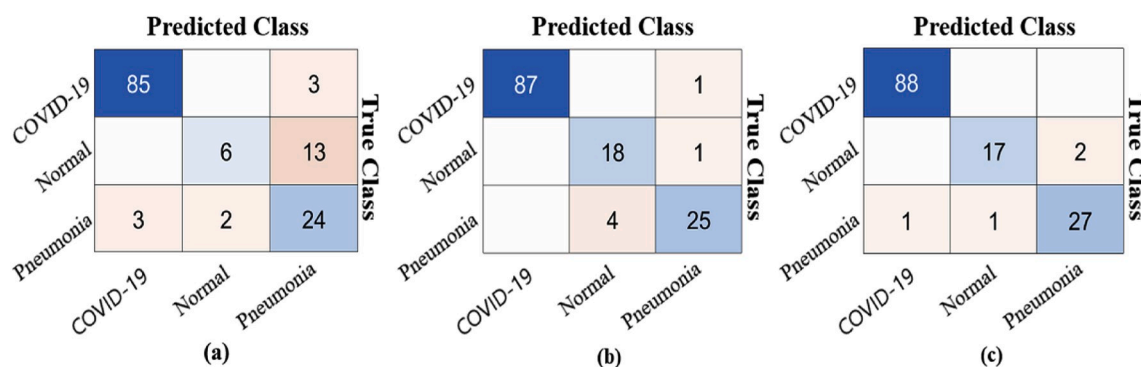


Fig. 9. Confusion matrices of the SqueezeNet model; (a) original dataset, (b) dataset restructured using the Fuzzy technique, (c) dataset combined using the Stacking technique.

analysis of this model are given in Table 6.

In the second step of this experiment, the k-fold cross-validation method was used for the stacked dataset, classified by using the SVM method. In the first step of the experiment, 30% of the dataset was used as test data. To confirm the validity of the results of the first step, the dataset was separated using a k-fold cross-validation method. For two CNN models, the k-fold value was adjusted to five. In the second step,

the overall accuracy rate achieved with the SqueezeNet model was 95.85%. In the first step of this model, the classification rate with the stacked dataset (with 30% test data) was 97.06%. In the analyzes performed in the second step, the SqueezeNet model produced stable results in both steps. In the second step, the overall accuracy rate was 96.28%, obtained from the MobileNetV2 model trained with the stacked dataset. In the first step of the MobileNetV2 model, the classification rate was

Table 5
Metric values of the confusion matrix obtained by the SqueezeNet model.

Model & Dataset Type	Classes	F-Scr. (%)	Se. (%)	Sp. (%)	Pre. (%)	Acc. (%)	Overall Acc. (%)
SqueezeNet & Original Data	COVID-19	96.59	96.59	90.91	96.59	95.04	84.56
	Normal	44.44	31.57	98.19	75	88.46	
	Pneumonia	69.56	82.75	85.04	60	84.56	
SqueezeNet & Structured dataset (Fuzzy Technique)	COVID-19	99.43	98.86	100	100	99.24	95.58
	Normal	87.80	94.73	96.55	81.82	96.29	
	Pneumonia	89.29	86.21	98.13	92.59	95.59	
SqueezeNet & Stacked dataset (Stacked Technique)	COVID-19	99.44	100	97.78	98.88	99.25	97.06
	Normal	91.89	89.47	99.14	94.44	97.78	
	Pneumonia	93.10	93.10	98.13	93.10	97.06	

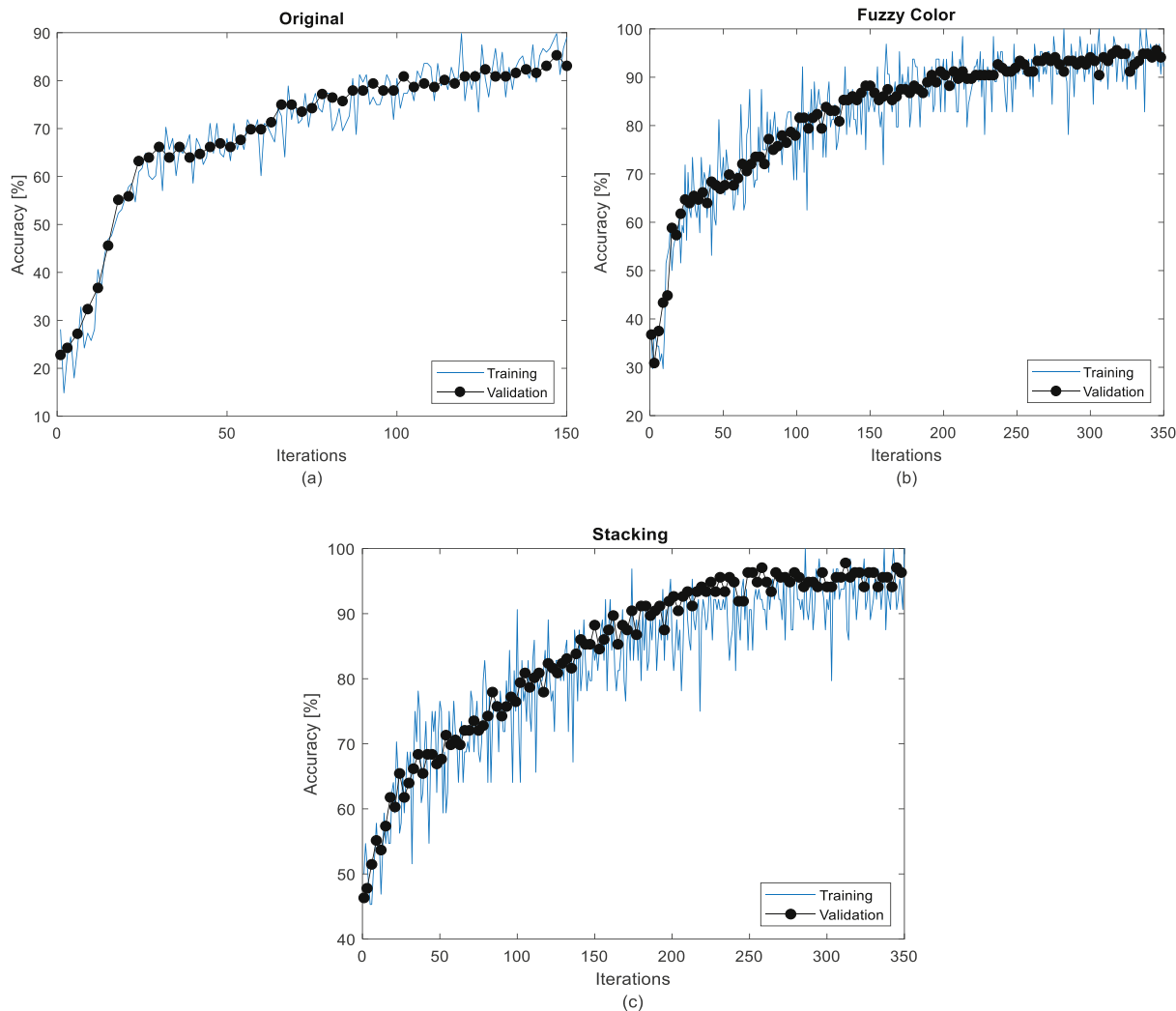


Fig. 10. Training and test success graphs of the MobileNetV2 model; (a) original dataset, (b) dataset restructured using the Fuzzy technique, (c) dataset combined using the Stacking technique.

97.06%, achieved with the stacked dataset (with 30% test data). Fig. 12 shows the confusion matrices of the analysis performed in the second step, and Table 7 gives the metric values obtained from the confusion matrices. As a result, all analyzes performed in the second step produce a stable result compared to the results obtained in the first step. The results obtained by the 5-fold cross-validation confirmed the reliability of the proposed approach compared to the results obtained from the previous step.

In the third step, the SMO algorithm was applied to the feature sets obtained from the stacked dataset trained by CNN models. The dataset

contains 1000 features, extracted by two CNN models. The dataset was obtained as a file with a '*.mat' extension from MATLAB. These feature sets were obtained from the Pool10 layer in the SqueezeNet model and the Logits layer in the MobileNetV2 model. Choosing efficient features with the SMO algorithm offers a total of 800 column numbers as we have set the maximum features selection to 800, and some of these column numbers are repeated using the SMO algorithm. The feature column numbers selected by CNN models are less than 800. Using an efficient optimization method with a few possible features, this step aimed to contribute to the classification success. The third step consists of four

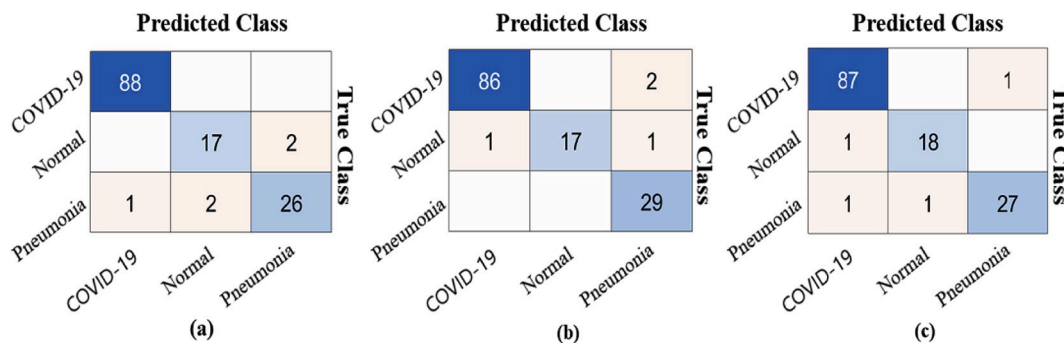


Fig. 11. Confusion matrices of the MobileNetV2 model; (a) original dataset, (b) dataset restructured using the Fuzzy technique, (c) dataset combined using the Stacking technique.

Table 6
Metric values of the confusion matrix obtained by the MobileNetV2 model.

Model & Dataset Type	Classes	F-Scr. (%)	Se. (%)	Sp. (%)	Pre. (%)	Acc. (%)	Overall Acc. (%)
MobileNetV2 & Original Data	COVID-19	99.43	100	97.72	98.87	99.24	96.32
	Normal	89.47	89.47	98.27	89.47	97.03	
	Pneumonia	91.22	89.65	98.13	92.85	96.32	
MobileNetV2 & Structured dataset (Fuzzy Technique)	COVID-19	98.29	97.73	97.87	98.85	97.78	97.05
	Normal	94.44	89.47	100	100	98.51	
	Pneumonia	95.08	100	97.17	90.63	97.78	
MobileNetV2 & Stacked dataset (Stacked Technique)	COVID-19	98.31	98.86	95.74	97.75	97.78	97.06
	Normal	94.74	94.74	99.13	94.74	98.51	
	Pneumonia	94.74	93.10	99.06	96.43	97.78	

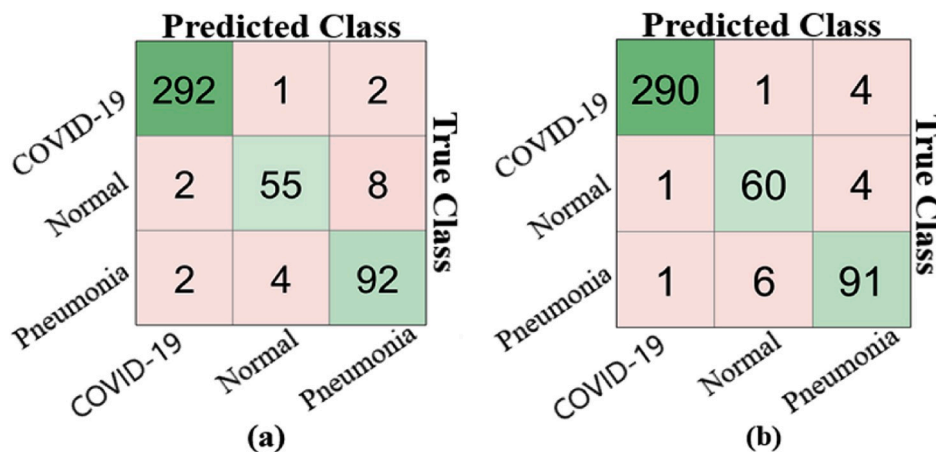


Fig. 12. Confusion matrices with the method of 5-fold cross-validation of stacked data; (a) using the SqueezeNet model, (b) using the MobileNetV2 model.

Table 7
Analysis results of the stacked dataset with the 5-fold cross-validation method.

Model & Dataset Type	Classes	F-Scr. (%)	Se. (%)	Sp. (%)	Pre. (%)	Acc. (%)	Overall Acc. (%)
SqueezeNet & Stacked dataset (Stacked Technique)	COVID-19	98.82	98.98	97.35	98.65	98.43	95.85
	Normal	88	84.62	98.71	96.70	96.70	
	Pneumonia	92	93.88	97.20	90.19	96.48	
MobileNetV2 & Stacked dataset (Stacked Technique)	COVID-19	98.81	98.31	98.69	99.32	98.44	96.28
	Normal	90.91	92.31	98.20	89.55	97.35	
	Pneumonia	92.39	92.86	97.77	91.92	96.71	

stages. In the first stage, the SMO algorithm was applied to the 1000-feature set, obtained by training the stacked dataset using the SqueezeNet model, and column numbers with efficient features were selected. The SqueezeNet model selected 694 efficient features from 1000 features, and the selected features were classified by the SVM method, which achieved an overall 97.81% accuracy rate. Also, the classification success in the detection of COVID-19 data with the SqueezeNet model was 100%. In the second stage, the SMO algorithm was applied to the 1000-feature set obtained from the stacked dataset with the MobileNetV2 model, and column numbers with efficient features were selected. The MobileNetV2 model selected 663 efficient features from 1000 features, and the efficient features were classified by the SVM method to achieve an overall 98.54% accuracy rate. Moreover, the classification success was 99.26% for the detection of COVID-19 data with the MobileNetV2 model. In the third stage, 694 efficient features obtained by the SMO algorithm using the SqueezeNet model were combined with the 663 efficient features obtained by the SMO algorithm and the MobileNetV2 model. The aim was to determine whether combining efficient features contributes to better classification performance. By combining two efficient features, a new dataset with 1357 features was obtained. Classified using the SVM method, 30% of the combined feature set was set as test data. The overall accuracy rate obtained from this classification was 99.27%. This result showed that combining selected features with the SMO algorithm contributes to the classification success. In the third stage, the accuracy rate in the classification of COVID-19 data was 100%, and the success rate was 99.27% in the classification of Normal chest images and Pneumonia chest images. In the fourth stage, the cross-validation method (k-fold = 5) was applied to

the combined feature set (dataset with 1357 features). This stage aimed to verify the validity of the success achieved in the third stage using the cross-validation method. The overall classification accuracy rate was 98.25% by using the SVM method. The result obtained in the last stage showed that the proposed approach is reliable and valid. In the fourth stage, the success rate was 99.34% in the classification of COVID-19 data, and the success rate was 98.68% in the classification of Normal chest images. The classification success rate was 98.47% in Pneumonia chest images. Fig. 13 shows the confusion matrices of the analysis performed in the third step of the experiment, and Table 8 gives the values of the metric parameters.

In the proposed approach, the contribution of the SMO algorithm was recorded for the improvement of classification performance. The codes and analysis of results of the SMO algorithm are available in the web address of the Open Source Code. Available in the web address is also feature sets obtained in the experiment and related source codes.

4. Discussion

The number of confirmed COVID-19 cases has exceeded millions worldwide with thousands of confirmed deaths. The World Health Organization has declared that COVID-19 is a global epidemic [37]. Using the proposed approach, we performed the detection of COVID-19 from the X-ray image data. We compared COVID-19 chest data with that of pneumonia and normal chest data since pneumonia is one of the symptoms of COVID-19. The major challenge we encountered is that the publication of COVID-19 images is still limited. Moreover, previous studies on the detection of COVID-19 using deep learning are

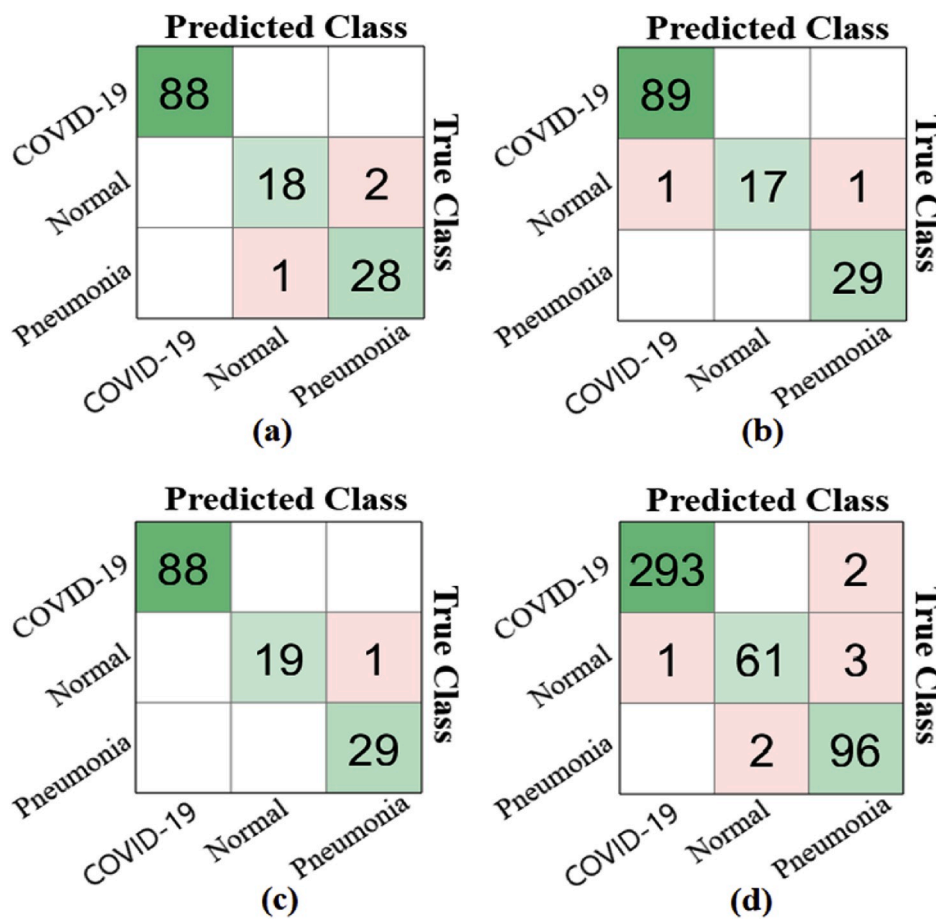


Fig. 13. Confusion matrices obtained using the SMO method; (a) with the SqueezeNet model, (b) with the MobileNetV2 model, (c) with combining the features from the SqueezeNet model and the MobileNetV2 model (30% test data), (d) with combining the features from the SqueezeNet model and the MobileNetV2 model (k fold value = 5).

Table 8
Metric values obtained using the SMO method.

Model / Dataset Type	Classes	Total of Features	Test Data (%)	F-Scr. (%)	Se. (%)	Sp. (%)	Pre. (%)	Acc. (%)	Overall Acc. (%)
SqueezeNet	COVID-19	694	30	100	100	100	100	100	97.81
	Normal			92.31	90	99.14	94.74	97.81	
	Pneumonia			94.91	96.55	98.15	93.33	97.81	
MobileNetV2	COVID-19	663	30	99.44	100	97.87	98.89	99.26	98.54
	Normal			94.44	89.47	100	100	98.54	
	Pneumonia			98.31	100	99.06	96.67	99.26	
SqueezeNet & MobileNetV2 (Combined Features Set)	COVID-19	1357	30	100	100	100	100	100	99.27
	Normal			97.43	95	100	100	99.27	
	Pneumonia			98.30	100	99.07	96.67	99.27	
	COVID-19	1357	k-fold (k = 5)	99.49	99.32	99.37	99.66	99.34	98.25
	Normal			95.31	93.85	99.49	96.82	98.68	
	Pneumonia			96.48	97.96	98.61	95.05	98.47	

non-existence. Hence, we fill the gap in the literature.

Although the limited dataset is used, this study contributes to the classification of the dataset, using the image preprocessing to determine the data classes. Other techniques can also be used instead of the Fuzzy Color technique. We paid attention to the similarity between the structured image and the original stack image. If we configured the image in a different format (such as resolution status or color pixel status), we would not have achieved the success achieved in this study.

The advantages of the proposed approach are as follows:

- It provides a 100% success rate in detecting the disease by examining the X-ray images of COVID-19 patients.
- The analysis can be carried out using AI, and the proposed approach can be integrated into portable smart devices (mobile phones, etc.)
- The deep learning models (MobileNetV2 and SqueezeNet) used in the proposed approach have fewer parameters compared to other deep models. This helps to gain speed and time performance. Besides, using the SMO algorithm, CNN models save time and speed during the process.
- It minimizes the interference in every image in the dataset and provides efficient features with stacking technique.

The disadvantages of the proposed approach are as follows:

- If the sizes of the input images in the dataset are different, a complete success may not be achieved. Irrespective of the resize parameter, it is still a challenge for the proposed approach to deal with very low-resolution images.
- In the Stacking technique, the resolution dimensions of the original images and the structured images must be the same.

We ensure that the number of pneumonia and normal chest images are close to that of COVID-19 chest images. Since we presume that the image classes found in an unbalanced number cannot contribute to the success of the model, we still achieved an overall 99.27% accuracy in the classification process.

In the proposed model, the end-to-end learning scheme has been exploited, which is one of the great advantages of CNN models. The pathologic patterns were detected and identified by using the activation maps that kept the discriminative features of the input data. In this manner, the tedious and labor-intensive feature extraction process was isolated; a highly sensitive decision tool was ensured.

5. Conclusion

People infected with COVID-19 are likely to suffer permanent damage in the lungs, which can later provoke death. This study aimed to distinguish people with damaged lungs owing to COVID-19 from normal individuals or pneumonia (not infected by COVID-19). The detection of

COVID-19 was carried out using deep learning models. Since it is important to detect COVID-19 that spread rapidly and globally, AI techniques are used to perform this accurately and quickly. One of the novelty aspects of the proposed approach is to apply the pre-processing steps to the images. When using pre-processing steps, more efficient features are extracted from the image data. With the stacking technique, each pixel of equivalent images is superimposed, and the pixels with low efficiency is increased. With the proposed approach, efficient features were extracted using the SMO algorithm. The model was aimed to produce faster and more accurate results. Another innovative aspect is that the feature sets obtained with SMO are combined to improve the classification performance. We also demonstrate the usability of our approach to smart mobile devices with the MobileNetV2 model as it can be analyzed on mobile devices without using any hospital devices. The 100% success was achieved in the classification of COVID-19 data, and 99.27% success was achieved in the classification of Normal and Pneumonia images.

In future studies, deep learning-based analyzes will be carried out using data images of other organs affected by the virus, in line with the views of COVID-19 specialists. We plan to develop a future approach using different structuring techniques to enhance the datasets. As the data related to the factors influencing the virus in human chemistry (e.g. blood group, RNA sequence, age, gender, etc.) are available, we will produce a solution-oriented analysis using AI.

Open Source Code

Information about Python and MATLAB software source codes, datasets, and related analysis results used in this study are given in this web link. https://github.com/mtogacar/COVID_19.

Funding

There is no funding source for this article.

Ethical approval

This article does not contain any data, or other information from studies or experimentation, with the involvement of human or animal subjects.

Declaration of competing interest

The authors declare that there is no conflict to interest related to this paper.

Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.cbi.2020.103805>.

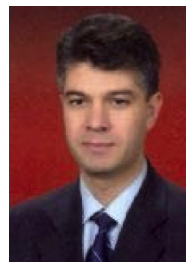
[org/10.1016/j.combiomed.2020.103805](https://doi.org/10.1016/j.combiomed.2020.103805).

References

- [1] C. Rothe, M. Schunk, P. Sothmann, G. Bretzel, G. Froeschl, C. Wallrauch, T. Zimmer, V. Thiel, C. Janke, W. Guggemos, M. Seilmaier, C. Drosten, P. Vollmar, K. Zwirgmaier, S. Zange, R. Wölfel, M. Hoelscher, Transmission of 2019-nCoV infection from an asymptomatic contact in Germany, *N. Engl. J. Med.* 382 (2020) 970–971, <https://doi.org/10.1056/nejmc2001468>.
- [2] T. Lancet, Editorial COVID-19 : too little , too late ? *Lancet* 395 (2020) 755, [https://doi.org/10.1016/S0140-6736\(20\)30522-5](https://doi.org/10.1016/S0140-6736(20)30522-5).
- [3] M.S. Razai, K. Doerholt, S. Ladhani, P. Oakeshott, Coronavirus Disease 2019 (Covid-19): a Guide for UK GPs, vol. 800, 2020, pp. 1–5, <https://doi.org/10.1136/bmj.m800>.
- [4] X. Peng, X. Xu, Y. Li, L. Cheng, X. Zhou, B. Ren, Transmission routes of 2019-nCoV and controls in dental practice, *Int. J. Oral Sci.* (2020) 1–6, <https://doi.org/10.1038/s41368-020-0075-9>.
- [5] M. Toğaçar, B. Ergen, Z. Cömert, Application of breast cancer diagnosis based on a combination of convolutional neural networks, ridge regression and linear discriminant analysis using invasive breast cancer images processed with autoencoders, *Med. Hypotheses* (2020) 109503, <https://doi.org/10.1016/j.mehy.2019.109503>.
- [6] X. Liu, Z. Deng, Y. Yang, Recent progress in semantic image segmentation, *Artif. Intell. Rev.* 52 (2019) 1089–1106, <https://doi.org/10.1007/s10462-018-9641-3>.
- [7] A.K. Jaiswal, P. Tiwari, S. Kumar, D. Gupta, A. Khanna, J.J.P.C. Rodrigues, Identifying pneumonia in chest X-rays: a deep learning approach, *Measurement* 145 (2019) 511–518, <https://doi.org/10.1016/j.measurement.2019.05.076>.
- [8] I.M. Baltruschat, H. Nickisch, M. Grass, T. Knopp, A. Saalbach, Comparison of deep learning approaches for multi-label chest X-ray classification, *Sci. Rep.* 9 (2019) 6381, <https://doi.org/10.1038/s41598-019-42294-8>.
- [9] S.S. Yadav, S.M. Jadhav, Deep convolutional neural network based medical image classification for disease diagnosis, *J. Big Data* 6 (2019) 113, <https://doi.org/10.1186/s40537-019-0276-2>.
- [10] R.H. Abiyev, M.K.S. Ma'aitah, Deep convolutional neural networks for chest diseases detection, *J. Healthc. Eng.* 2018 (2018) 4168538, <https://doi.org/10.1155/2018/4168538>.
- [11] O. Stephen, M. Sain, U.J. Maduh, D.-U. Jeong, An efficient deep learning approach to pneumonia classification in healthcare, *J. Healthc. Eng.* 2019 (2019) 4180949, <https://doi.org/10.1155/2019/4180949>.
- [12] V. Chouhan, S.K. Singh, A. Khamparia, D. Gupta, P. Tiwari, C. Moreira, R. Damaševićius, V.H.C. de Albuquerque, A novel transfer learning based approach for pneumonia detection in chest X-ray images, *Appl. Sci.* 10 (2020), <https://doi.org/10.3390/app10020559>.
- [13] J.P. Cohen, COVID-19 Chest X-Ray dataset or CT dataset, GitHub. <https://github.com/ieee8023/covid-chestxray-dataset>, 2020 accessed March 10, 2020.
- [14] T. Rahman, M. Chowdhury, A. Khandakar, COVID-19 Radiography Database, Kaggle, 2020. <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database/data#>. accessed April 20, 2020.
- [15] M. Hosseiny, S. Kooraki, A. Gholamrezanezhad, S. Reddy, L. Myers, Radiology perspective of coronavirus disease 2019 (COVID-19): lessons from severe acute respiratory syndrome and Middle East respiratory syndrome, *Am. J. Roentgenol.* (2020) 1–5, <https://doi.org/10.2214/AJR.20.22969>.
- [16] A. Ahmed, Pneumonia Sample X-Rays, GitHub, 2019. <https://www.kaggle.com/ahmedali2019/pneumonia-sample-xrays>. accessed March 10, 2020.
- [17] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.C. Chen, MobileNetV2: inverted residuals and linear bottlenecks, *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn.* (2018) 4510–4520, <https://doi.org/10.1109/cvpr.2018.00474>.
- [18] Google Ai Blog, MobileNetV2: the next generation of on-device computer vision networks, n.d. <https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html>. accessed January 8, 2020.
- [19] M. Toğaçar, B. Ergen, Z. Cömert, BrainMRNet: brain tumor detection using magnetic resonance images with a novel convolutional neural network model, *Med. Hypotheses* 134 (2020) 109531, <https://doi.org/10.1016/j.mehy.2019.109531>.
- [20] G. Zhang, T. Lei, Y. Cui, P. Jiang, A dual-path and lightweight convolutional neural network for high-resolution aerial image segmentation, *ISPRS Int. J. Geo-Inf.* 8 (2019), <https://doi.org/10.3390/ijgi8120582>.
- [21] G. Fu, P. Sun, W. Zhu, J. Yang, Y. Cao, M.Y. Yang, Y. Cao, A deep-learning-based approach for fast and robust steel surface defects classification, *Optic Laser. Eng.* 121 (2019) 397–405, <https://doi.org/10.1016/j.optlaseng.2019.05.005>.
- [22] H.J. Lee, I. Ullah, W. Wan, Y. Gao, Z. Fang, Real-time vehicle make and model recognition with the residual SqueezeNet architecture, *Sensors (Basel)* 19 (2019), <https://doi.org/10.3390/s19050982>.
- [23] I. Sharif, D. Chaudhuri, A multiseed-based SVM classification technique for training sample reduction, *Turk. J. Electr. Eng. Comput. Sci.* 27 (2019) 595–604, <https://doi.org/10.3906/elk-1801-157>.
- [24] Y. Wang, W. Yu, Z. Fang, Multiple kernel-based SVM classification of hyperspectral images by combining spectral, spatial, and semantic information, *Rem. Sens.* 12 (2020), <https://doi.org/10.3390/rs12010120>.
- [25] P. Netrapalli, Stochastic gradient descent and its variants in machine learning, *J. Indian Inst. Sci.* 99 (2019) 201–213, <https://doi.org/10.1007/s41745-019-0098-4>.
- [26] M. Awad, R. Khanna, in: M. Awad, R. Khanna (Eds.), *Support Vector Machines for Classification BT - Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, Apress, Berkeley, CA, 2015, pp. 39–66, https://doi.org/10.1007/978-1-4302-5990-9_3.
- [27] Ü. Doğan, T. Glasmachers, C. Igel, A unified view on multi-class support vector classification, *J. Mach. Learn. Res.* 17 (2016) 1–32.
- [28] T. Bardak, S. Bardak, Prediction of wood density by using red-green-blue (RGB) color and fuzzy logic techniques, *J. Polytech.* 20 (2017) 979–984, <https://doi.org/10.2339/politeknik.369132>.
- [29] J. Arnal, L. Súcar, Hybrid filter based on fuzzy techniques for mixed noise reduction in color images, *Appl. Sci.* 10 (2020), <https://doi.org/10.3390/app10010243>.
- [30] J.M. Soto-Hidalgo, D. Sánchez, J. Chamorro-Martínez, P.M. Martínez-Jiménez, Color comparison in fuzzy color spaces, *Fuzzy Set Syst.* (2019), <https://doi.org/10.1016/j.fss.2019.09.013>.
- [31] V. Patrascu, Fuzzy color image enhancement algorithm, Github (2019), <https://doi.org/10.13140/2.1.3014.6562>.
- [32] N.A. Elleboudy, H.M. Ezz Eldin, S.M.S. Azab, Focus stacking technique in identification of forensically important Chrysomya species (Diptera: calliphoridae), *Egypt. J. Forensic Sci.* 6 (2016) 235–239, <https://doi.org/10.1016/j.ejfs.2016.06.001>.
- [33] Y. Gingold, Image stack: simple code to load and process image stacks. <https://github.com/yig/imagestack>, 2019 accessed March 11, 2020.
- [34] S. Balochian, H. Balochian, Social mimic optimization algorithm and engineering applications, *Expert Syst. Appl.* 134 (2019) 178–191, <https://doi.org/10.1016/j.eswa.2019.05.035>.
- [35] E. Cengil, A. Çınar, A new approach for image classification: convolutional neural network, *Eur. J. Teach. Educ.* 6 (2016) 96–103.
- [36] Z. Cömert, Fusing fine-tuned deep features for recognizing different tympanic membranes, *Biocybern. Biomed. Eng.* 40 (2020) 40–51, <https://doi.org/10.1016/j.bbe.2019.11.001>.
- [37] Coronavirus disease, n.d. <https://www.who.int/emergencies/diseases/novel-coronavirus-2019>, 2019. accessed March 13, 2020



Mesut Toğaçar is currently working as a lecturer at Firat University, Technical Sciences Vocational High School. In 2008, he received his undergraduate degree from Firat University, Department of Electronics and Computer Education. In 2014, he received his bachelor's degree in Computer Engineering from Firat University. In 2015, he received his Master's degree from Firat University, Department of Electronic Computer Education. Since 2017, he is a PhD candidate at Firat University, Department of Computer Engineering.



Burhan Ergen is currently Asst. Prof. in Department of Computer Engineering at Firat University. He received his BS degree in Electronics Engineering from Karadeniz Technical University in 1993. He received his master's degree from Karadeniz Technical University in 1996 and his doctorate degree from Firat University in 2004. He is currently working at Firat University.



Zafer Cömert is currently Asst. Prof. in Department of Software Engineering at Samsun University. He received his BSc degree in Electronics and Computer Education from Firat University in 2008. He also received his MSc degree in Computer and Instructional Technologies from Firat University in 2012. He has finished Ph.D. work on the classification of cardiocography data with machine learning techniques in Department of Computer Engineering at İnönü University in 2017.