A novel comparative study for detection of Covid-19 on CT lung images using texture analysis, machine learning, and deep learning methods

Huseyin Yasar1 · Murat Ceylan2

Received: 19 April 2020 / Revised: 5 September 2020 / Accepted: 16 September 2020 / Published online: 6 October 2020 © Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract
The Covid-19 virus outbreak that emerged in China at the end of 2019 caused a huge and devastating effect worldwide. In patients with severe symptoms of the disease, pneumonia develops due to Covid-19 virus. This causes intense involvement and damage in lungs. Although the emergence of the disease occurred a short time ago, many literature studies have been carried out in which these effects of the disease on the lungs were revealed by the help of lung CT imaging. In this study, 1,396 lung CT images in total (386 Covid-19 and 1,010 Non-Covid-19) were subjected to automatic classification. In this study, Convolutional Neural Network (CNN), one of the deep learning methods, was used which suggested automatic classification of CT images of lungs for early diagnosis of Covid-19 disease. In addition, k-Nearest Neighbors (k-NN) and Support Vector Machine (SVM) was used to compare the classification successes of deep learning with machine learning. Within the scope of the study, a 23-layer CNN architecture was designed and used as a classifier. Also, training and testing processes were performed for Alexnet and Mobilenetv2 CNN architectures as well. The classification results were also calculated for the case of increasing the number of images used in training for the first 23-layer CNN architecture by 5, 10, and 20 times using data augmentation methods. To reveal the effect of the change in the number of images in the training and test clusters on the results, two different training and testing processes, 2-fold and 10-fold cross-validation, were performed and the results of the study were calculated. As a result, thanks to these detailed calculations performed within the scope of the study, a comprehensive comparison of the success of the texture analysis method, machine learning, and deep learning methods in Covid-19 classification from CT images was made. The highest mean sensitivity, specificity, accuracy, F-1 score, and AUC values obtained as a result of the study were 0.9197, 0.9891, 0.9473, 0.9058, 0.9888; respectively for 2-fold cross-validation, and they were 0.9404, 0.9901, 0.9599, 0.9284, 0.9903; respectively for 10-fold cross-validation.

Keywords Covid-19 · Convolutional neural networks (CNN) · Deep learning · Lung CT classification · Machine learning · Texture analysis methods
1 Introduction

In December 2019, an outbreak occurred in Wuhan of China’s Hubei province, caused by a new corona virus which is of zootonic origin and affecting acute respiratory tract in general [46]. With the spread of the virus in question in the following days, the disease began to seriously affect all countries in the world. The World Health Organization announced in March 2020 that the disease became a global pandemic. In addition, the World Health Organization named this new epidemic in question “Covid-19”. Common symptoms of the disease are fever, cough, shortness of breath, muscle pain, and weakness [36]. The disease has quite severe and negative effects on the lungs in general. In this context, many literature studies have been realised in a short time in which these effects of the disease on lungs were shown by the help of lung CT imaging. These studies reveal that besides clinical symptoms, blood and biochemical tests, lung CT imaging is an important diagnostic tool for the diagnosis of the disease as well.

In a study conducted by Qin et al. [33], a clinical examination images of lung CT belonging to four cases, two men and two women with Covid-19 disease, was performed. The results of the study reveal that lung lesions due to Covid-19 show a high degree of involvement in patients with pneumonia disease. In the study carried out by Albarello et al. [2], the change in the images related to chest X-Rays and CT of lungs belonging to 2 Covid-19 cases in Italy in the disease process was investigated. In the context of the study, it was evaluated by the authors that monitoring the deterioration of lungs by radiological images is an important alternative for the early diagnosis of the disease, considering the clinical findings. In the study conducted by Lin et al. [24], the change in CT images of lungs belonging to a 61-year-old male Covid-19 patient in accordance with the progress in the disease was examined. In the scope of the study, it is reported that there is an increase in lung involvement depending on the progress of the disease. In a similar study by Li et al. [21], CT images of lung belonging to 5 Covid-19 patients, whose ages range from 10 months to 6 years were evaluated clinically. As a result of the study, it was reported that 2 of these patients had no signs of disease on CT images of lungs but 3 of them showed significant abnormalities.

In the study conducted by Xu et al. [41], CT images of lungs belonging to 50 Covid-19 patients in total, of whose 9 are in mild, 28 moderate, 10 severe, and 3 critical level, were evaluated clinically. In the examinations carried out within the scope of the study, it was reported that no changes in radiological images occurred in 9 patients, while symmetrical lesions developed in 26 patients and asymmetrical lesions in 15 patients. In addition, in the discussion part of the study, it was evaluated that repeated CT scanning is a useful method for monitoring disease progress and timely treatment of Covid-19. In the study performed by Xia et al. [39], CT images of 20 Covid-19 diagnosed pediatric patients were analyzed. The results of the study reveal that all patients have sub-pleural lesions. A similar study by Chen et al. [10] was carried out by analyzing the clinical data history of 9 women who were diagnosed with Covid-19 and were also pregnant. As a result of the study, it is stated that CT images of lungs have high diagnostic value in Covid-19 diagnosis. A study of clinical data of patients diagnosed with Covid-19 was performed by Huang et al. [18]. It was reported that lung abnormalities were detected in 40 of 41 Covid-19 patients examined within the scope of the study and that there was bilateral involvement.

In the study performed by Hu et al. [16], the CT images of lung belonging to 2 Covid-19 patients were evaluated. In the scope of the study, although the symptoms of the disease decreased after two days of treatment, this reduction and recovery showed inconsistency with the CT images of lungs. In the study conducted by Liu et al. [25], clinical evaluation
of CT images of lungs belonging to 73 Covid-19 cases of various severities was performed. The results of the study show that all patients except the patient group of 8 percent who undergo the disease at the level of mild pneumonia had abnormal CT images of lungs. In the study conducted by Xu et al. [42], the clinical data histories of 90 Covid-19 patients were examined. From the results of the study, it was reported that Covid-19 patients had multiple patchy ground glass opacities on CT images. A similar study was carried out by Pan et al. [31], using the clinical data of 63 Covid-19 patients and abnormalities were detected in CT images of lungs belonging to the patients. As for the study conducted by Shen et al. [37], the lesion levels in CT images of lungs belonging to 44 Covid-19 patients were labeled by radiologists and computers respectively. The study results show that computerized labeling is a reliable alternative method in detecting the severity and distribution of pneumonia due to Covid-19 disease. In a study conducted by Li et al. [22], a clinical evaluation of CT images of lungs taken in the course of pneumonia due to Covid-19 disease was performed. In the study, attention was drawn to the importance of CT imaging of lungs in understanding the effect and progress of the Covid-19 disease. Clinical symptoms and CT images of lungs belonging to a 54-year-old male patient with Covid-19 disease in South Korea were investigated by Lim et al. [23]. A similar study was conducted by Cheng et al. [11] for the first Covid-19 case in Taiwan.

In a review study conducted by Long and Ehrenfeld [26], it is emphasized that using artificial intelligence methods to reduce the effects of Covid-19 outbreak crisis is an essential requirement. In this context, some applications regarding the automatic classification of Covid-19 and Non-Covid-19 CT images with artificial intelligence methods were implemented. Detailed information about the methods used in these studies, number of images, training-test methods and study results are given in Table 1.

In this study, Convolutional Neural Network (CNN), one of the deep learning methods, was used which suggested automatic classification of CT images of lungs for early diagnosis of Covid-19 disease. In addition, k-Nearest Neighbors (k-NN) and Support Vector Machine (SVM) were used to compare the classification successes of deep learning with machine learning. Within the scope of the study, besides the results obtained when the images are used directly with the SVM, k-NN, and CNN classifiers, the results obtained by using Gray-Level Co-Occurrence Matrix (GLCM), Local Binary Pattern (LBP), and Local Entropy (LE) as a pre-process were found out as well. In addition, the success of the proposed method was tested separately in case that all the image data were combined as double, triple, and quadruple.

Within the scope of the study, a 23-layer CNN architecture was designed and used as a classifier. In addition, training and testing processes were performed for Alexnet and Mobilenetv2 CNN architectures as well. The classification results were also calculated for the case of increasing the number of images used in training for the first 23-layer CNN architecture by 5, 10, and 20 times using data augmentation methods. In order to reveal the effect of the change in the number of images in the training and test clusters on the results, two different training and testing processes, 2-fold and 10-fold cross-validation, were performed and the results of the study were calculated. As a result, thanks to these detailed calculations performed within the scope of the study, a comprehensive comparison of the success of the texture analysis method, machine learning, and deep learning methods in Covid-19 classification from CT images was made. The results of the study indicate that analyzing CT images of lungs with the help of deep learning methods in diagnosing Covid-19 disease will speed up the diagnosis and significantly reduce the burden on healthcare workers.
| Study                | Year | No. of images (Covid-19 and Non-Covid-19) | Methods                                                                 | Test methods                  | Results                                                                 |
|---------------------|------|------------------------------------------|------------------------------------------------------------------------|-------------------------------|------------------------------------------------------------------------|
| Han et al. [14]     | 2020 | 460 images (230 Covid-19 and 230 Non-Covid-19) | C3D, DeCoVNet and AD3D-MIL Algorithm                                   | 60% Train-20% Validation-20% Test | Sen: 0.968-0.979; Spe: X; Acc: 0.968-0.979; F-1 Score: 0.968-0.979; AUC: 0.982-0.990; Time: X |
| Ardakani et al. [4] | 2020 | 1020 images (510 Covid-19 and 510 Non-Covid-19) | Pre-processing (Cropped / Input Image Size: 60x60) and Transfer Learning with Convolutional Neural Networks (AlexNet, VGG-16, VGG-19, SqueezeNet, GoogleNet, MobileNet-V2, ResNet-18, ResNet-50, ResNet-101, and Xception) | 80% Train-20% Test | Sen: 0.7843-1.000; Spe: 0.6863-1.000; Acc: 0.7892-0.9951; F-1 Score: X; AUC: 0.894-0.994; Time: X |
| Jaiswal et al. [19] | 2020 | 2492 images (1262 Covid-19 and 1230 Non-Covid-19) | Transfer Learning with Convolutional Neural Networks (VGG16, Inception ResNet, Resnet 152V2, DenseNet201) | 68% Train-17% Validation-15% Test | Sen: 0.9206-0.9735; Spe: 0.8972-0.9621; Acc: 0.909-0.9625; F-1 Score: 0.9109-0.9629; AUC: 0.97; Time: X |
| Pathak et al. [32]  | 2020 | 852 images (413 Covid-19 and 439 Non-Covid-19) | Transfer Learning with Convolutional Neural Networks (Transfer from ResNet-50 Network to a New CNN Architecture) | 10-fold and 50% Train-10% Validation-40% Test | Sen: 0.9146; Spe: 0.9478; Acc: 0.9302; F-1 Score: X; AUC: X; Time: X |
| Ouyang et al. [30]  | 2020 | 4982 images (3389 Covid-19 and 1593 Non-Covid-19) | Convolutional Neural Network (3D Resnet34) and Uniform Sampling, Size-balanced Sampling, Dual-Sampling | 5-fold and Train: 2186 (1094 Covid-19 and 1092 Non-Covid-19) Test: 2796 (2295 Covid-19 and 501 Non-Covid-19) | Sen: 0.869; Spe: 0.901; Acc: 0.875; F-1 Score: 0.82; AUC: 0.944; Time: X |
| Sakagianni et al. [34] | 2020 | 746 images (349 Covid-19 and 397 Non-Covid-19) | Auto-ML platform (Google AutoML Cloud Vision) | Train: 596 (279 Covid-19 and 317 Non-Covid-19) Validation: 73 (34 Covid-19 and 39 Non-Covid-19) Test: 77 (36 Covid-19 and 41 Non-Covid-19) | Sen: 0.8831; Spe: X; Acc: X; F-1 Score: 0.8831; AUC: X; Time: X |
| Hu et al. [17]      | 2020 | 300 images (150 Covid-19 and 150 Non-Covid-19) | Weakly Supervised Deep Learning | 5-fold | Sen: 0.833; Spe: 0.956; Acc: 0.906; F-1 Score: X; AUC: 0.943; Time: X |
2 Material and method

2.1 Used data

The CT images of lungs belonging to Covid-19 patients used in the study were taken from two data set created and accumulated in the form of metadata by Cohen et al. [13] and Zhao et al. [45] who then made it available to public through Github for open access. In this context, 386 CT images of lungs in total were used in the study. The sizes of the images in question are between $115 \times 98$ and $2024 \times 1523$ and vary widely. First of all, in order to clarify the area of interest on the images, framing was carried out so as to include the lung region. After this process, the dimensions of all the images were rearranged as $448 \times 448$. In the scope of the study, 1,010 Non-Covid-19 CT images of lungs were also used. The images in question were similarly taken from, LIDC-IDRI public access research database [1, 5, 6, 12], a previously published data set. The processes applied to the Covid-19 images were also applied to the Non-Covid-19 ones.

2.2 Local Binary Pattern (LBP)

Local Binary Pattern (LBP) was first introduced by Ojala et al. [27]. This method is often used to reveal local spatial structures. LBP is a sequential comparison process of a center pixel with its neighboring pixel values. So it is a non-parametric method. The first example of this comparison was made for a $3 \times 3$ square operator. Then, operations were realized for operators of different sizes too. In (1), there is a mathematical representation of this operation.

$$LBP_{p,r} = \sum_{n=0}^{p-1} s(x_{r,n} - x_{0,0})2^n$$

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

As can be seen from the (1), threshold measurement is made by realizing a comparison with neighbors of the central pixel. As a result of the threshold measurement, a LBP code with a total of 256 combinations comes out. This code is then mathematically converted to a local number, the new pixel value. A sample LBP application is shown in Fig. 1.

In case LBP is applied to the images, it is seen that the image sizes decrease. For example, when an LBP operation with a radius 2 is applied to an image of $224 \times 224$, an image of $220 \times 220$ is obtained. The reason for this is that this operation cannot be applied to the pixels in the starting and ending rows and columns. The diameter of the LBP used in this study is 2. To prevent this size reduction from causing any problem in the operations mentioned above,
the dimensions of the image obtained after LBP operations were restored to 448 × 448. The use of the LBP process in the study enable us to have new images from the original ones that reflect local features. Thus, the total image feature depth was increased.

2.3 Local Entropy (LE)

Entropy can be explained as the measure of uncertainty or randomness in an image [43]. The global entropy of an image can be mathematically calculated with (2). In this formula, \( p_i \) is obtained from the normalized histogram of the image and is called the probability associated with the pixel value \( i \). If local histograms are used instead of global histograms, local entropy is calculated. Local entropy is an effective feature extraction method used in texture analysis. CT lung images were included in the study to reveal the texture properties.

\[
H = - \sum_{i=0}^{255} p_i \log_2 p_i \tag{2}
\]

2.4 Gray-Level Co-Occurrence Matrix (GLCM)

Gray-Level Co-Occurrence Matrix (GLCM) is used to reveal how often pairs of pixels with a specified value and shift occur in an image [40]. The GLCM conjugate of an image can be mathematically calculated with (3).

\[
C_{\Delta x, \Delta y}(i, j) = \sum_{x=1}^{n} \sum_{y=1}^{m} \begin{cases} 1, & I(x, y) = i \text{ and } I(x + \Delta x, y + \Delta y) = j \\ 0, & Otherwise \end{cases} \tag{3}
\]

In this equation, \( I \) denote the image in size \( n \times m \), of which the GLCM is calculated, \( i \) and \( j \) are the pixel values; \( x \) and \( y \) denote the spatial position of the image and \( C \) is the co-occurrence matrix. The size of the matrix \( C \) is \( p \times p \), with \( p \) the number of different pixels in \( I \) image. For this reason, as in LBP, the re-arrangement of the dimensions is realized after the GLCM process. The GLCM, being a method frequently used for texture analysis in image processing applications, was included in the study to reveal different image features of CT lung images and to increase the variety of features.

In Fig. 2, the original versions of a Covid-19 and Non-Covid-19 image and the versions obtained by applying LBP, LE, and GLCM are given.

2.5 Data augmentation

Within the scope of the study, the effects of the increase in the number of images used in the training set of the CNN classifier on the study results were also examined. For this purpose, data were augmented using classical data augmentation methods. The number of 1.396 CT lung images (386 Covid-19 and 1.010 Non-Covid-19) was augmented by 5 times, 10 times, and 20 times using conventional data augmentation methods. In this context, data augmentation was realized by using contrast changing, brightness changing, distortion, and noise adding methods on the original images. In some steps of performing these processes, the image augmentation tools created by the works of Bloice et al. [7–9] were used.

First, the pixel values related to the original image size being multiplied by 0.8 and 0.6 respectively and the contrast being changed, the second and third images were obtained. The brightness change was made by increasing the value of each pixel of the second and third images by 7 and the fourth and fifth images were created. After these processes, the number of images amounted to 5 times the previous number.
Secondly, the original images were eroded by the distortion process. Later, the processes performed in the first stage were also applied to the eroded images. At the end of the second stage, the number of images amounted to 10 times the number of images obtained in the first stage.

A salt and pepper noise of a density of 0.03 was added to the original image and the first nine images obtained from this image, and the number of images was increased to 20. The selection of the type of noise to be added and the parameter sizes used in the data augmentation processes was made by taking into consideration the literature studies [29] and [28] having successful results. In this context, it is possible to choose different parameter sizes and noise types and to use other data augmentation methods.

By using the data augmentation methods, the amount of LBP, LE, and GLCM images as well as the original images were augmented by 5 times, 10 times, and 20 times. At this stage, data augmentation processes applied to the original images were performed using these images, instead of the original image as the source image.

### 2.6 Support Vector Machine (SVM)

Support Vector Machine (SVM) can be broadly described as a supervised learning approach. It is based on statistical learning theory. The foundations of SVM were demonstrated by the studies realized by Vapnik and Chervonenkis [38]. SVM can effectively perform linear and nonlinear classifications. Two-class SVM was used within the scope of the study. Radial Basis Function (RBF) was preferred as a kernel function in SVM. While the results were standardized, other parameters were used as the default (Matlab 2019a).

### 2.7 k-Nearest Neighbors (k-NN)

k-Nearest Neighbors (k-NN) is a non-parametric classification and regression method [3]. When classifying is made using k-NN, the object to be classified is voted on by its neighbors...
and assigned to the most common class among its closest neighbors. Two-class k-NN was used within the scope of the study. While the neighbor number of k-NN was set 21, other parameters were used as the default (Matlab 2019a).

One of the main objectives of the study is to compare the success of machine learning with deep learning methods in Covid-19 classification. For this purpose, SVM and k-NN, which are basic machine learning methods, were included in the study as classifiers.

2.8 Convolutional Neural Network (CNN)

Deep learning is realized using too many layers in the realization of learning process. Convolutional Neural Network (CNN) is the most frequently used model of deep learning. This model has come into use widely, especially in image processing applications in recent years. CNN consist of some layers such as convolutional layer, activation function, pooling and fully connected layer. Convolutional layers are often designed to follow each other and enable us to obtain feature patterns from low-level features of images to high-level features [15]. Activation functions in CNN architecture can be defined as the functions that bring incoming inputs to a certain range or accept some of the input values while eliminating some of them again. Pooling layers, on the other hand, let the size of feature matrices to be reduced, through sampling. As for fully connected layer, it is the layer where the classification process is performed according to the features obtained through convolution, activation function and pooling. This layer works like a classic artificial neural network. Before the classification process, the conversion of feature matrices into feature vectors, that is, flattening process is performed. In this context, Fig. 3 shows the general architecture of the CNN classifier.

Within the scope of the study, a CNN architecture consisting of 23 layers in total was designed. Table 2 shows the information and parametric features of the layers of this CNN architecture. In programming processes, Matlab 2019a program was used and the function names and parameters of the layers used in the program were written directly in the layer name and parameters sections. The reason why there are four different dimensions in the first layer, the image input layer, is that experiments were made for multiple and different sized input images within the scope of the study.

In addition to the 23-layer CNN architecture presented within the scope of the study, two more CNN architectures that were previously presented in the literature and proved to be successful in many studies, were used. In this context, Alexnet [20] and Mobilenetv2 [35] CNN architectures were also included in the study. While only the architectures of these CNN models were transferred, the previously trained coefficients were not transferred. Alexnet and Mobilenetv2 were trained from the beginning with the images used in the study.

![Fig. 3 General architecture of the CNN classifier](Image)
Table 2 Features of CNN architecture constructed within the scope of the study

| Layer number | Layer name                  | Layer parameters (Matlab)                      |
|--------------|-----------------------------|------------------------------------------------|
| 1            | imageInputLayer             | [448 448 1], [448 448 2], [448 448 3], and [448 448 4] |
| 2            | convolution2dLayer          | (3,4, 'Padding', 'same')                       |
| 3            | batchNormalizationLayer     | default                                        |
| 4            | reluLayer                   | default                                        |
| 5            | maxPooling2dLayer           | (2, 'Stride', 2)                               |
| 6            | convolution2dLayer          | (3,8, 'Padding', 'same')                       |
| 7            | batchNormalizationLayer     | default                                        |
| 8            | reluLayer                   | default                                        |
| 9            | maxPooling2dLayer           | (2, 'Stride', 2)                               |
| 10           | convolution2dLayer          | (3,16, 'Padding', 'same')                      |
| 11           | batchNormalizationLayer     | default                                        |
| 12           | reluLayer                   | default                                        |
| 13           | maxPooling2dLayer           | (2, 'Stride', 2)                               |
| 14           | convolution2dLayer          | (3,32, 'Padding', 'same')                      |
| 15           | batchNormalizationLayer     | default                                        |
| 16           | reluLayer                   | default                                        |
| 17           | maxPooling2dLayer           | (2, 'Stride', 2)                               |
| 18           | convolution2dLayer          | (3,64, 'Padding', 'same')                      |
| 19           | batchNormalizationLayer     | default                                        |
| 20           | reluLayer                   | default                                        |
| 21           | fullyConnectedLayer         | 2                                              |
| 22           | softmaxLayer                | default                                        |
| 23           | classificationLayer         | default                                        |

Alexnet is a CNN architecture that consists of 25 layers and has a serially arranged structure. While the input image dimensions of the original architecture of the Alexnet model are 227×227×3 the fully connected layer output size is 1000. In other words, a classification with 1000 classes is performed at the output of this architecture. Mobilenetv2, on the other hand, has an architecture consisting of 155 layers and 164 connection points that is, arranged both serially and parallelly. While the initial image dimensions of the original architecture of Mobilenetv2 model are 224×224×3, the size of the fully connected layer output size is 1000.

Some modifications need to be made on Alexnet and Mobilenetv2 architectures for them to be used in the study. In this context, the number of lines and columns in the input image was preserved as the values in the original architecture. However, the size of the third dimension of the input data was rearranged as 1, 2, 3, and 4 according to the requirements of the study. In this context, the new input image sizes of the modified Alexnet architecture are 227×227×1, 227×227×2, 227×227×3, 227×227×4 and the new input image sizes of the modified Mobilenetv2 architecture are 224×224×1, 224×224×2 is 224×224×3, 224×224×4. Within the scope of the study, since a two-class classification, Covid-19 and Non-Covid-19, was performed, the fully connected layer output size was re-adjusted as 2 in both CNN architectures. For this reason, the dimensions of the images used in the operations.
realized using Alexnet and Mobilenetv2 are different from the first CNN architecture and machine learning classifiers. With these architectures, the images were re-sized using the imresize (Matlab 2019a) function. This issue was also taken into consideration additionally, in the interpretation of CPU run-times within the scope of the study.

2.9 The evaluation criteria for the classification results

In this study, besides the parameters: TP, TN, FN, FP and the dependent variables such as sensitivity (SEN), specificity (SPE), F-1 Score (F-1), and accuracy (ACC) which were obtained by mathematical processing of the values of the above-mentioned parameters were used in the evaluation of the results. Within the scope of the study, Receiver Operating Characteristic (ROC) analysis was made and the area (AUC) sizes under the ROC curve were compared.

In this context, TP is the number of times that the actually patient data is labeled as patients as a result of classification as well. FP, on the other hand, is the number of times that non-patient data is labeled as patient in the same way. TN is the number of times that non-patient data is labeled as not patient as a result of the classification. FN, on the other hand, is the labeling of the patient data as non-patient in the same way. Sensitivity (SEN), specificity (SPE), F-1 Score (F-1), and accuracy (ACC) values calculated using these parameters are defined mathematically between (4)–(7). Within the scope of the study, SEN, SPE, F-1, and ACC values were calculated for the threshold (cut-off) value to be 0.5.

\[ SEN = \frac{TP}{TP + FN} \]  
\[ SPE = \frac{TN}{TN + FP} \]  
\[ ACC = \frac{(TP + TN)}{(TP + TN + FP + FN)} \]  
\[ F - 1 = \frac{(2 \times TP)}{(2 \times TP + FP + FN)} \]

ROC analysis examines the change of sensitivity (SEN) (y-axis) relative to the discrimination threshold value, that is, the cut-off value [44], in relation to the precision that is the complement of the specificity to one (1-SPE) (x-axis) in the classification processes performed in two groups. In this context, the area under the curve that reflects the change in question is called AUC. This field approaching to 1 indicates that the classification goes to perfection, while approaching 0 indicates that the classification deteriorates.

3 Experiments and results

3.1 Experiments

Within the scope of the study, machine learning-based and deep learning-based artificial intelligence applications were implemented for the automatic diagnosis of CT images of lung for early diagnosis of Covid-19 disease. In this study, a total of 1,396 lung CT images were used, 386 of which were belonging to patients diagnosed with Covid-19 and 1,010 to Non-Covid-19 subjects.

In the study, first of all, framing process was carried out to include the lung region in order to clarify the interests of the related images. Since the images used are of very different sizes, the images were rearranged in size and the image dimensions were set to 448×448. Later, the images in question were recorded in gray-scale format and in 8 bit depth in png format. In this context, it is possible to name these processes as pre-process applications.
In the second part of the study, new images were created by applying LBP, LE, and GLCM to the original images of Covid-19 patients and Non-Covid-19 patients and recorded in the same size, format and depth as the original images. In this context, since there was a decrease in image sizes after LBP and GLCM application, those images were rearranged in 448×448 dimensions. In this way, new images containing feature patterns of the relevant images were created and the features of texture were enriched. Within the scope of the study, the effects of the increase in the number of images used in the training set of the CNN classifier on the study results were also examined. In this stage of the study, data augmentation was performed by the help of changing contrast, distortion, and brightness as well as noise-adding methods described earlier. This data augmentation process was performed for the images applied LBP, LE, and GLCM as well as the original images. As in the first stage, these images were recorded in gray-scale format and 8 bit depth in png format. At the end of the second phase, the amount of data was augmented 5 times, 10 times, and 20 times, including first images on which were based derivation.

In the third stage of the study, first, the original images with LBP, LE, and GLCM images separately and secondly in the form of double, triple and quadruple combinations including the original image in each group were subjected to Covid-19 and Non-covid-19 classification using SVM and k-NN classifier. In order to observe the effects of the change in the training-test image number balance on the results, two different training-test processes, that is, 2-fold cross-validation and 10-fold cross-validation, were performed and the results were calculated.

In the fourth part of the study, a 23-layer CNN architecture, previously described, was developed and this architecture was used in all the experiments. In this section, the image groups created in the previous step were classified using this 23-layer CNN architecture. As in the previous stage, 2-fold cross-validation and 10-fold cross-validation were used for training-test processes in classification. Each experiment was repeated 5 times in itself in order to let the results reach stability due to the random assignment of some initial weights and parameters used in the structure of CNN. The average of the test results obtained as a result of five repetitions was given in the scope of the study.

In the fifth stage of the study, the study results were calculated for the case of increasing the number of images and using them in 23-layer CNN training with the help of the data augmentation techniques detailed previously. In this context, training-testing processes were realized to increase by three different amounts: 5, 10, and 20 times. In these training-test processes, only 2-fold cross-validation was performed due to the length of processing times.

The original images were divided into two, so as to be they are in the same group as the images derived from them. That is, when the original image is in the training group, it is prevented that a derived image obtained from this image is in the test group or vice versa. While the training process was performed by augmenting the data, the testing process was performed only for the original images.

In the sixth part of the study, the training and testing processes were realized using the Alexnet and Mobilenetv2 CNN architecture which were detailed previously and of which the modification processes were performed. At this stage, training and testing processes were realized using 2-fold and 10-fold cross-validation as in the third and fourth stages. However, unlike these stages, instead of 448×448, the dimensions of the input images were used as 227×227 for Alexnet and 224×224 for Mobilenetv2.

Within the scope of the study, the times (CPU time) needed to be able to classify an image were also measured. The experiments of this study were carried out using MATLAB 2019 (a) software running on 64 GB RAM and Intel(R) Xeon (R) CPU E5-2680 2.7 GHz (32 CPUs).
3.2 Results

Within the scope of the study, first the Covid-19/Non-Covid-19 classification was performed using the SVM classifier. The dimensions of the input images for this experiment were 448×448×1 for Original, GLCM, LBP, and LE; 448×448×2 for Original+GLCM, Original+LBP, and Original+LE; 448×448×3 for Original+GLCM+LBP, Original+GLCM+LE, and Original+LBP+LE; 448×448×4 for Original+GLCM+LBP+LE. These image matrices were given as input to SVM matrices after being converted to image vectors. The study results obtained in this context were given in Table 3 for 2-fold cross-validation and Table 4 for 10-fold cross-validation. The highest parameter values in Table 3 and Table 4 are stated in bold. This marking is also made in the following tables.

Within the scope of the study, secondly, the Covid-19/Non-Covid-19 classification was performed using the k-NN classifier. The dimensions of the input images for this experiment are the same as those used for the SVM classifier. These image matrices were given as input to k-NN matrices after being converted to image vectors. The study results obtained in this context were given in Table 5 for 2-fold cross-validation and Table 6 for 10-fold cross-validation.

Within the scope of the study, thirdly, the Covid-19/Non-Covid-19 classification was performed using the 23-layer CNN classifier. The dimensions of the input images for this experiment were 448×448×1 for Original, GLCM, LBP, and LE; 448×448×2 for Original+GLCM, Original+LBP, and Original+LE; 448×448×3 for Original+GLCM+LBP, Original+GLCM+LE, and Original+LBP+LE; 448×448×4 for Original+GLCM+LBP+LE. The study results obtained in this context were given in Table 7 for 2-fold cross-validation and Table 8 for 10-fold cross-validation. Each experiment was repeated 5 times in itself in order to let the results reach stability due to the random assignment of some initial weights and parameters used in the structure of CNN. The average of the test results obtained in five repetitions were given within the scope of the study. This is true for all experiments performed after this test.

Within the scope of the study, fourthly, the Covid-19/Non-Covid-19 classification was performed using the 23-layer CNN classifier and data augmentation (×5, ×10, and ×20). The dimensions of the input images for this experiment were 448×448×1 for Original,

| Method                      | TP   | FN  | TN  | FP  | SEN  | SPE  | ACC  | F-1  | AUC  | CPU Time |
|-----------------------------|------|-----|-----|-----|------|------|------|------|------|----------|
| Original                    | 342  | 49  | 968 | 42  | 0.8860| 0.9584| 0.9384| 0.8883| 0.9777| 1.5857   |
| GLCM                        | 341  | 45  | 981 | 29  | 0.8834| 0.9713| 0.9470| 0.9021| 0.9815| 1.6846   |
| LBP                         | 335  | 51  | 971 | 39  | 0.8679| 0.9614| 0.9355| 0.8816| 0.9744| 3.6905   |
| LE                          | 340  | 46  | 965 | 45  | 0.8808| 0.9554| 0.9348| 0.8820| 0.9777| 3.4185   |
| Original+GLCM               | 342  | 44  | 969 | 41  | 0.8860| 0.9594| 0.9391| 0.8895| 0.9800| 3.1503   |
| Original+LBP                | 335  | 51  | 971 | 39  | 0.8679| 0.9614| 0.9355| 0.8816| 0.9744| 3.6905   |
| Original+LE                 | 340  | 46  | 965 | 45  | 0.8808| 0.9554| 0.9348| 0.8820| 0.9777| 3.4185   |
| Original+GLCM+LBP           | 337  | 49  | 971 | 39  | 0.8731| 0.9614| 0.9370| 0.8845| 0.9784| 5.2163   |
| Original+GLCM+LE            | 344  | 42  | 970 | 40  | **0.8912**| 0.9604| 0.9413| 0.8935| 0.9801| 4.9756   |
| Original+LBP+LE             | 335  | 51  | 969 | 41  | 0.8679| 0.9594| 0.9341| 0.8793| 0.9768| 5.6864   |
| Original+GLCM+LBP+LE        | 341  | 45  | 971 | 39  | 0.8834| 0.9614| 0.9398| 0.8903| 0.9795| 7.0295   |
GLCM, LBP, and LE; $448 \times 448 \times 2$ for Original+GLCM, Original+LBP, and Original+LE; $448 \times 448 \times 3$ for Original+GLCM+LBP, Original+GLCM+LE, and Original+LBP+LE; $448 \times 448 \times 4$ for Original+GLCM+LBP+LE. In this experiment, training and testing processes were performed using only 2-fold cross-validation due to the increase in the time to obtain results in parallel with the increase in the training set. The results obtained in the experiments were given in Table 9 for increasing the number of training images by 5 times, in Table 10 for increasing by 10 times, and in Table 11 for 20 times.

Within the scope of the study, fifthly, the Covid-19/Non-Covid-19 classification was performed using the Alexnet CNN classifier. The dimensions of the input images for this experiment were $448 \times 448 \times 1$ for Original, GLCM, LBP, and LE; $448 \times 448 \times 2$ for Original+GLCM, Original+LBP, and Original+LE; $448 \times 448 \times 3$ for Original+GLCM+LBP, Original+GLCM+LE, and Original+LBP+LE; $448 \times 448 \times 4$ for Original+GLCM+LBP+LE. The study results obtained in this context were given in Table 12 for 2-fold cross-validation and Table 13 for 10-fold cross-validation.

### Table 4 Results obtained by using SVM classifier for 10-fold cross-validation

| Method         | TP  | FN  | TN  | FP  | SEN | SPE | ACC  | F-1  | AUC  | CPU Time |
|----------------|-----|-----|-----|-----|-----|-----|------|------|------|----------|
| Original       | 354 | 32  | 976 | 34  | 0.9171 | 0.9663 | 0.9527 | 0.9147 | 0.9853 | 9.2526    |
| GLCM           | 349 | 37  | 989 | 21  | 0.9041 | 0.9792 | 0.9585 | 0.9233 | 0.9874 | 9.6001    |
| LBP            | 94  | 292 | 1010| 0   | 0.2435 | 1.0000 | 0.9708 | 0.3917 | 0.8554 | 25.4466   |
| LE             | 321 | 65  | 991 | 19  | 0.8316 | 0.9812 | 0.9398 | 0.8843 | 0.9751 | 13.0530   |
| Original+GLCM  | 351 | 35  | 978 | 32  | 0.9093 | 0.9683 | 0.9520 | 0.9129 | 0.9866 | 18.3973   |
| Original+LBP   | 344 | 42  | 977 | 33  | 0.8912 | 0.9673 | 0.9463 | 0.9017 | 0.9840 | 21.4024   |
| Original+LE    | 346 | 40  | 977 | 33  | 0.8964 | 0.9673 | 0.9477 | 0.9046 | 0.9849 | 19.8091   |
| Original+GLCM+LBP | 346 | 40  | 977 | 33  | 0.8964 | 0.9673 | 0.9477 | 0.9046 | 0.9860 | 30.1233   |
| Original+GLCM+LE | 348 | 38  | 978 | 32  | 0.9016 | 0.9683 | 0.9499 | 0.9086 | 0.9866 | 28.7369   |
| Original+LBP+LE | 342 | 44  | 980 | 30  | 0.8860 | 0.9703 | 0.9470 | 0.9024 | 0.9853 | 32.7798   |
| Original+GLCM+LBP+LE | 346 | 40  | 981 | 29  | 0.8964 | 0.9713 | 0.9506 | 0.9093 | 0.9868 | 41.9181   |

### Table 5 Results obtained by using k-NN classifier for 2-fold cross-validation

| Method         | TP  | FN  | TN  | FP  | SEN | SPE | ACC  | F-1  | AUC  | CPU Time |
|----------------|-----|-----|-----|-----|-----|-----|------|------|------|----------|
| Original       | 300 | 86  | 975 | 35  | 0.7772 | 0.9653 | 0.9133 | 0.8322 | 0.9800 | 2.4588    |
| GLCM           | 267 | 119 | 997 | 13  | 0.6917 | 0.9871 | 0.9054 | 0.8018 | 0.9872 | 2.5106    |
| LBP            | 116 | 270 | 929 | 81  | 0.3005 | 0.9198 | 0.7486 | 0.3979 | 0.6987 | 2.5268    |
| LE             | 160 | 226 | 1010| 0   | 0.4145 | 1.0000 | 0.8381 | 0.5861 | 0.9806 | 2.5013    |
| Original+GLCM  | 292 | 94  | 990 | 20  | 0.7565 | 0.9802 | 0.9183 | 0.8367 | 0.9862 | 5.4448    |
| Original+LBP   | 298 | 88  | 979 | 31  | 0.7720 | 0.9693 | 0.9148 | 0.8336 | 0.9830 | 5.4867    |
| Original+LE    | 291 | 95  | 992 | 18  | 0.7539 | 0.9822 | 0.9191 | 0.8374 | 0.9867 | 5.4546    |
| Original+GLCM+LBP | 286 | 100 | 993 | 17  | 0.7409 | 0.9832 | 0.9162 | 0.8302 | 0.9868 | 7.9491    |
| Original+GLCM+LE | 282 | 104 | 999 | 11  | 0.7306 | 0.9891 | 0.9176 | 0.8306 | 0.9885 | 8.1147    |
| Original+LBP+LE | 292 | 94  | 993 | 17  | 0.7565 | 0.9832 | 0.9205 | 0.8403 | 0.9875 | 8.3712    |
| Original+GLCM+LBP+LE | 282 | 104 | 998 | 12  | 0.7306 | 0.9881 | 0.9169 | 0.8294 | 0.9888 | 11.0160   |
Within the scope of the study, sixthly, the Covid-19/Non-Covid-19 classification was performed using the Mobilenetv2 CNN classifier. The dimensions of the input images for this experiment were 448×448×1 for Original, GLCM, LBP, and LE; 448×448×2 for Original+GLCM, Original+LBP, and Original+LE; 448×448×3 for Original+GLCM+LBP, Original+GLCM+LE, and Original+LBP+LE; 448×448×4 for Original+GLCM+LBP+LE. The study results obtained in this context were given in Table 14 for 2-fold cross-validation and Table 15 for 10-fold cross-validation.

### 4 Conclusion

Within the scope of the study, the results of the study are given with a total of 13 Tables (Tables 3–15) in which there are experimental results for 11 different input image in each
Table 8  Results obtained by using 23-layer CNN classifier for 10-fold cross-validation

| Method        | TP     | FN     | TN     | FP     | SEN   | SPE   | ACC   | F-1   | AUC   | CPU   |
|---------------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|
| Original      | 357.0  | 29.0   | 978.4  | 31.6   | 0.9249| 0.9687| 0.9566| 0.9217| 0.9845| 1.1252|
| GLCM          | 349.4  | 36.6   | 986.6  | 23.4   | 0.9052| 0.9768| 0.9570| 0.9209| 0.9902| 1.1235|
| LBP           | 322.4  | 63.6   | 996.4  | 13.6   | 0.8352| 0.9447| 0.9865| 0.9447| 0.9814| 1.1242|
| LE            | 340.4  | 45.6   | 991.8  | 18.2   | 0.8819| 0.9820| 0.9543| 0.9143| 0.9849| 1.1236|
| Original+GLCM | 354.0  | 32.0   | 974.2  | 35.8   | 0.9171| 0.9646| 0.9514| 0.9126| 0.9789| 1.7136|
| Original+LBP  | 349.8  | 36.2   | 979.2  | 30.8   | 0.9062| 0.9695| 0.9520| 0.9126| 0.9854| 1.7178|
| Original+LE   | 349.4  | 36.6   | 977.0  | 33.0   | 0.9052| 0.9673| 0.9501| 0.9094| 0.9818| 1.7174|
| Original+GLCM+LBP | 350.4 | 35.6   | 980.8  | 29.2   | 0.9078| 0.9711| 0.9536| 0.9153| 0.9873| 2.3014|
| Original+GLCM+LE | 350.6 | 35.4   | 980.2  | 29.8   | 0.9083| 0.9705| 0.9533| 0.9149| 0.9843| 2.3003|
| Original+LBP+LE | 349.4 | 36.6   | 983.4  | 26.6   | 0.9052| 0.9737| 0.9547| 0.9171| 0.9873| 2.3034|
| Original+GLCM+LBP+LE | 348.6 | 37.4   | 981.4  | 28.6   | 0.9031| 0.9717| 0.9527| 0.9136| 0.9849| 2.8945|

The evaluation of the best results obtained in these experiments has carried out in this section.

When Tables 3 and 4, which includes the results obtained by using the SVM classifier, are examined within the scope of the study, the highest mean sensitivity, specificity, accuracy, F-1 score, and AUC values were found to be 0.8912, 1.0000, 0.9470, 0.9021, 0.9815; respectively for 2-fold cross-validation, 0.9171, 1.0000, 0.9585, 0.9233, 0.9874; respectively for 10-fold cross-validation. It was observed that although the highest mean specificity value was obtained as 1.0000, there were serious worsenings in sensitivity, accuracy, F-1 score, and AUC values in experiments where this value was obtained. The second highest specificity obtained using the SVM classifier is 0.9713 for 2-fold cross-validation and 0.9812 for 10-fold cross-validation. In obtaining these highest average values mentioned, it is seen that the combinations of input images such as GLCM and Original+GLCM+LE for 2-fold

Table 9  Results obtained by using 23-layer CNN classifier and data augmentation (5 times) for 2-fold cross-validation

| Method                    | TP    | FN    | TN    | FP    | SEN   | SPE   | ACC   | F-1   | AUC   | CPU   |
|---------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Original                  | 333.6 | 52.4  | 934.8 | 75.2  | 0.8642| 0.9255| 0.9086| 0.8397| 0.9545| 0.6499|
| GLCM                      | 330.0 | 50.0  | 951.8 | 58.2  | 0.8549| 0.9442| 0.9182| 0.8356| 0.9598| 0.6438|
| LBP                       | 275.8 | 110.2 | 977.2 | 32.8  | 0.7145| 0.9675| 0.8976| 0.7940| 0.9580| 0.6448|
| LE                        | 326.6 | 59.4  | 974.6 | 35.4  | 0.8461| 0.9650| 0.9321| 0.8734| 0.9703| 0.6479|
| Original+GLCM             | 339.0 | 47.0  | 940.0 | 70.0  | 0.8782| 0.9307| 0.9162| 0.8528| 0.9604| 0.9850|
| Original+LBP              | 338.0 | 48.0  | 953.4 | 56.6  | 0.8756| 0.9440| 0.9251| 0.8659| 0.9685| 0.9868|
| Original+LE               | 338.0 | 48.0  | 949.0 | 61.0  | 0.8756| 0.9396| 0.9219| 0.8610| 0.9696| 0.9858|
| Original+GLCM+LBP         | 338.8 | 47.2  | 951.2 | 58.8  | 0.8777| 0.9418| 0.9241| 0.8647| 0.9668| 1.3134|
| Original+GLCM+LE          | 331.6 | 54.4  | 945.8 | 64.2  | 0.8591| 0.9364| 0.9150| 0.8484| 0.9603| 1.3129|
| Original+LBP+LE           | 341.8 | 44.2  | 959.4 | 50.6  | 0.8855| 0.9499| 0.9321| 0.8782| 0.9736| 1.3140|
| Original+GLCM+LBP+LE      | 339.0 | 47.0  | 957.2 | 52.8  | 0.8782| 0.9477| 0.9285| 0.8717| 0.9691| 1.6378|
Table 10 Results obtained by using 23-layer CNN classifier and data augmentation (10 times) for 2-fold cross-validation

| Method          | TP    | FN    | TN    | FP    | SEN   | SPE   | ACC   | F-1   | AUC   | CPU   |
|-----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Original        | 334.2 | 51.8  | 938.6 | 71.4  | 0.8658| 0.9293| 0.9117| 0.8444| 0.9565| 1.2964|
| GLCM            | 327.6 | 58.4  | 956.4 | 53.6  | 0.8487| 0.9469| 0.9198| 0.8542| 0.9617| 1.2972|
| LBP             | 291.6 | 94.4  | 979.6 | 30.4  | 0.7554| 0.9699| 0.9106| 0.8235| 0.9597| 1.2998|
| LE              | 330.6 | 55.4  | 978.6 | 31.4  | 0.8565| 0.9689| 0.9378| 0.8840| 0.9709| 1.3060|
| Original+GLCM   | 336.2 | 49.8  | 948.6 | 61.4  | 0.8710| 0.9392| 0.9203| 0.8580| 0.9625| 1.9531|
| Original+LBP    | 335.0 | 51.0  | 957.0 | 53.0  | 0.8679| 0.9475| 0.9255| 0.8656| 0.9695| 1.9568|
| Original+LE     | 334.6 | 51.4  | 951.6 | 58.4  | 0.8686| 0.9422| 0.9213| 0.8591| 0.9636| 1.9528|
| Original+GLCM+LBP| 342.0 | 44.0  | 958.2 | 51.8  | 0.8860| 0.9487| 0.9314| 0.8771| 0.9723| 2.6482|
| Original+GLCM+LE| 345.8 | 40.2  | 948.6 | 61.4  | 0.8959| 0.9392| 0.9272| 0.8719| 0.9707| 2.6338|
| Original+LBP+LE | 339.8 | 46.2  | 966.2 | 43.8  | 0.8803| 0.9566| 0.9355| 0.8830| 0.9724| 2.6160|
| Original+GLCM+LBP+LE | 339.2 | 46.8  | 974.4 | 35.6  | 0.8788| 0.9648| 0.9410| 0.8917| 0.9780| 3.5341|

When Tables 5 and 6, which includes the results obtained by using the k-NN classifier, are examined within the scope of the study, the highest mean sensitivity, specificity, accuracy, F-1 score, and AUC values were found to be 0.7772, 1.0000, 0.9205, 0.8403, 0.9888; respectively for 2-fold cross-validation, 0.8057, 0.9980, 0.9284, 0.8567, 0.9903; respectively for 10-fold cross-validation. As is with the SVM classifier, although the highest mean specificity values were obtained for the k-NN classifier as 1.0000 and 0.9980 as well, serious worsenings occur in the sensitivity, accuracy, F-1 score, and AUC values.

Table 11 Results obtained by using 23-layer CNN classifier and data augmentation (20 times) for 2-fold cross-validation

| Method          | TP    | FN    | TN    | FP    | SEN   | SPE   | ACC   | F-1   | AUC   | CPU   |
|-----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Original        | 329.6 | 56.4  | 939.2 | 70.8  | 0.8539| 0.9299| 0.9089| 0.8382| 0.9566| 2.6773|
| GLCM            | 332.6 | 53.4  | 934.0 | 76.0  | 0.8617| 0.9248| 0.9073| 0.8372| 0.9566| 2.6038|
| LBP             | 299.8 | 86.2  | 971.0 | 39.0  | 0.7767| 0.9614| 0.9103| 0.8272| 0.9634| 2.6043|
| LE              | 333.0 | 53.0  | 974.6 | 35.4  | 0.8627| 0.9650| 0.9367| 0.8828| 0.9731| 2.6017|
| Original+GLCM   | 336.6 | 49.4  | 945.2 | 64.8  | 0.8720| 0.9358| 0.9182| 0.8549| 0.9648| 4.1130|
| Original+LBP    | 335.6 | 50.4  | 960.0 | 50.0  | 0.8694| 0.9505| 0.9281| 0.8699| 0.9700| 3.9790|
| Original+LE     | 340.0 | 46.0  | 959.0 | 51.0  | 0.8808| 0.9495| 0.9305| 0.8752| 0.9668| 4.0007|
| Original+GLCM+LBP| 335.8 | 50.2  | 965.0 | 45.0  | 0.8699| 0.9554| 0.9318| 0.8759| 0.9745| 12.6129|
| Original+GLCM+LE| 338.2 | 47.8  | 966.4 | 43.6  | 0.8762| 0.9568| 0.9345| 0.8809| 0.9743| 12.6575|
| Original+LBP+LE | 342.4 | 43.6  | 961.4 | 48.6  | 0.8870| 0.9519| 0.9340| 0.8813| 0.9753| 11.4009|
| Original+GLCM+LBP+LE | 341.8 | 44.2  | 960.4 | 49.6  | 0.8855| 0.9509| 0.9328| 0.8793| 0.9835| 16.3373|
### Table 12 Results obtained by using Alexnet CNN classifier for 2-fold cross-validation

| Method            | TP   | FN   | TN   | FP   | SEN  | SPE  | ACC  | F-1  | AUC  | CPU Time |
|-------------------|------|------|------|------|------|------|------|------|------|----------|
| Original          | 334.4| 51.6 | 932.2| 77.8 | 0.8663| 0.9230| 0.9073| 0.8381| 0.9518| 0.0557   |
| GLCM              | 336.6| 49.4 | 929.2| 80.8 | 0.8720| 0.9200| 0.9067| 0.8381| 0.9558| 0.0516   |
| LBP               | 316.8| 69.2 | 944.6| 65.4 | 0.8207| 0.9352| 0.9036| 0.8237| 0.9482| 0.0517   |
| LE                | 348.6| 37.4 | 971.6| 38.4 | 0.9031|       |       |       | 0.9620| 0.9457   |
| Original+GLCM     | 342.2| 43.8 | 934.4| 75.6 | 0.8865| 0.9251| 0.9145| 0.8514| 0.9557| 0.0704   |
| Original+LBP      | 338.4| 47.6 | 924.2| 85.8 | 0.8767| 0.9150| 0.9044| 0.8355| 0.9480| 0.0700   |
| Original+LE       | 346.4| 39.6 | 928.8| 81.2 | 0.8974| 0.9196| 0.9135| 0.8518| 0.9586| 0.0687   |
| Original+GLCM+LBP | 346.0| 40.0 | 918.8| 91.2 | 0.8964| 0.9097| 0.9060| 0.8406| 0.9545| 0.0885   |
| Original+GLCM+LE  | 348.0| 38.0 | 936.4| 73.6 | 0.9016| 0.9271| 0.9201| 0.8622| 0.9603| 0.0882   |
| Original+LBP+LE   | 349.0| 37.0 | 920.2| 89.8 |       | 0.9011| 0.9092| 0.8466| 0.9538| 0.0881   |
| Original+GLCM+LBP+LE | 343.0| 43.0 | 914.0| 96.0 | 0.8886| 0.9050| 0.9004| 0.8318| 0.9470| 0.1062   |

Values in the experiments where these values were obtained. The second highest specificity obtained using the k-NN classifier is 0.9891 for 2-fold cross-validation and 0.9901 for 10-fold cross-validation. In obtaining these highest average values mentioned, it is seen that the combinations of input images such as Original, Original+GLCM, Original+LBP, and Original+GLCM+LBP+LE for 2-fold cross-validation, the Original, Original+LBP, and Original+GLCM+LBP+LE were used for 10-fold cross-validation. When the processing times are examined, it is understood that results can be obtained for an image at times ranging from 2.4588 to 11.0160 seconds for 2-fold cross-validation and from 3.9970 to 22.7328 seconds for 10-fold cross-validation, in direct proportion to the image data sizes.

When Tables 7 and 8, which includes the results obtained by using the 23-layer CNN classifier, are examined within the scope of the study, the highest mean sensitivity, specificity, accuracy, F-1 score, and AUC values were found to be 0.8865, 0.9756, 0.9394,

### Table 13 Results obtained by using Alexnet CNN classifier for 10-fold cross-validation

| Method            | TP   | FN   | TN   | FP   | SEN  | SPE  | ACC  | F-1  | AUC  | CPU Time |
|-------------------|------|------|------|------|------|------|------|------|------|----------|
| Original          | 350.6| 35.4 | 965.6| 44.4 | 0.9083| 0.9560| 0.9428| 0.8978| 0.9738| 0.4070   |
| GLCM              | 345.2| 40.8 | 974.4| 35.6 | 0.8943| 0.9648| 0.9453| 0.9003| 0.9774| 0.4026   |
| LBP               | 338.2| 47.8 | 970.0| 40.0 | 0.8762| 0.9604| 0.9371| 0.8851| 0.9731| 0.4027   |
| LE                | 347.2| 38.8 | 991.2| 18.8 | 0.8995|       |       |       | 0.9814| 0.9587   |
| Original+GLCM     | 353.2| 32.8 | 970.0| 40.0 | 0.9150| 0.9604| 0.9479| 0.9066| 0.9738| 0.5537   |
| Original+LBP      | 356.0| 30.0 | 964.6| 45.4 | 0.9223| 0.9550| 0.9460| 0.9042| 0.9740| 0.5550   |
| Original+LE       | 353.2| 32.8 | 971.4| 38.6 | 0.9150| 0.9618| 0.9489| 0.9082| 0.9788| 0.5542   |
| Original+GLCM+LBP | 356.4| 29.6 | 959.6| 50.4 | 0.9233| 0.9501| 0.9427| 0.8991| 0.9723| 0.7096   |
| Original+GLCM+LE  | 351.6| 34.4 | 967.0| 43.0 | 0.9109| 0.9574| 0.9446| 0.9008| 0.9729| 0.7082   |
| Original+LBP+LE   | 359.0| 27.0 | 962.0| 48.0 |       | 0.9301| 0.9525| 0.9463| 0.9054| 0.9720   |
| Original+GLCM+LBP+LE | 354.8| 31.2 | 956.8| 53.2 | 0.9192| 0.9473| 0.9395| 0.8937| 0.9676| 0.8628   |
Results obtained by using Mobilenetv2 CNN classifier for 2-fold cross-validation

| Method          | TP    | FN    | TN    | FP    | SEN   | SPE   | ACC   | F-1   | AUC   | CPU |
|-----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-----|
| Original        | 355.0 | 31.0  | 964.8 | 45.2  | **0.9197** | 0.9552 | 0.9454 | 0.9031 | 0.9759 | 0.7444 |
| GLCM            | 336.4 | 49.6  | 967.8 | 42.2  | 0.8715 | 0.9582 | 0.9342 | 0.8799 | 0.9775 | 0.7382 |
| LBP             | 262.4 | 123.6 | 952.6 | 57.4  | 0.6798 | 0.9432 | 0.8703 | 0.7431 | 0.9217 | 0.7395 |
| LE              | 270.0 | 116.0 | 978.8 | 31.2  | 0.6995 | 0.9497 | 0.9282 | 0.8695 | 0.9628 | 0.7622 |
| Original+GLCM   | 353.6 | 32.4  | 968.8 | 41.2  | 0.9161 | 0.9592 | **0.9473** | **0.9058** | **0.9783** | 0.7595 |
| Original+LBP    | 319.6 | 66.4  | 960.2 | 49.8  | 0.8280 | 0.9507 | 0.9282 | 0.8695 | 0.9628 | 0.7516 |
| Original+LE     | 333.8 | 52.2  | 962.0 | 48.0  | 0.8648 | 0.9525 | 0.9282 | 0.8695 | 0.9628 | 0.7622 |
| Original+GLCM+LBP | 326.2 | 59.8  | 959.2 | 50.8  | 0.8451 | 0.9497 | 0.9208 | 0.8551 | 0.9693 | 0.7743 |
| Original+GLCM+LE | 337.4 | 48.6  | 965.0 | 45.0  | 0.8741 | 0.9554 | 0.9330 | 0.8782 | 0.9627 | 0.7790 |
| Original+LBP+LE | 323.2 | 62.8  | 972.6 | 37.4  | 0.8373 | 0.9630 | 0.9282 | 0.8656 | 0.9666 | 0.7725 |
| Original+GLCM+LBP+LE | 323.4 | 62.6  | 967.0 | 43.0  | 0.8378 | 0.9574 | 0.9244 | 0.8596 | 0.9643 | 0.7987 |

Results obtained by using Mobilenetv2 CNN classifier for 10-fold cross-validation

| Method            | TP    | FN    | TN    | FP    | SEN   | SPE   | ACC   | F-1   | AUC   | CPU |
|-------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-----|
| Original          | 363.0 | 23.0  | 977.0 | 33.0  | **0.9404** | 0.9673 | **0.9599** | **0.9284** | 0.9833 | 6,6215 |
| GLCM              | 352.2 | 33.8  | 983.4 | 26.6  | 0.9124 | 0.9737 | 0.9567 | 0.9210 | 0.9860 | 6,6210 |
| LBP               | 285.8 | 100.2 | 967.0 | 43.0  | 0.7404 | 0.9574 | 0.8974 | 0.7996 | 0.9452 | 6,5895 |
| LE                | 307.8 | 78.2  | 988.0 | 22.0  | 0.7974 | **0.9782** | 0.9282 | 0.8600 | 0.9677 | 6,6118 |
| Original+GLCM     | 360.0 | 26.0  | 974.0 | 36.0  | 0.9326 | 0.9644 | 0.9556 | 0.9207 | 0.9858 | 6,7781 |
| Original+LBP      | 349.4 | 36.6  | 976.0 | 34.0  | 0.9052 | 0.9663 | 0.9494 | 0.9082 | 0.9794 | 6,8004 |
| Original+LE       | 348.4 | 37.6  | 972.6 | 37.4  | 0.9026 | 0.9630 | 0.9463 | 0.9028 | 0.9789 | 6,7878 |
| Original+GLCM+LBP | 351.6 | 34.4  | 977.2 | 32.8  | 0.9109 | 0.9675 | 0.9519 | 0.9127 | 0.9841 | 6,9414 |
| Original+GLCM+LE  | 358.2 | 27.8  | 975.2 | 34.8  | 0.9280 | 0.9655 | 0.9552 | 0.9196 | 0.9816 | 6,9408 |
| Original+LBP+LE   | 341.6 | 44.4  | 972.6 | 37.4  | 0.8850 | 0.9630 | 0.9414 | 0.8931 | 0.9745 | 6,9451 |
| Original+GLCM+LBP+LE | 346.4 | 39.6  | 976.2 | 33.8  | 0.8974 | 0.9665 | 0.9474 | 0.9041 | 0.9838 | 7,1508 |

0.8871, 0.9727; respectively for 2-fold cross-validation, 0.9249, 0.9865, 0.9570, 0.9217, 0.9902; respectively for 10-fold cross-validation. In obtaining these highest average values mentioned, it is seen that the combinations of input images such as GLCM, LBP, LE, and Original+GLCM+LBP+LE for 2-fold cross-validation, the Original, GLCM, and LBP were used for 10-fold cross-validation. When the processing times are examined, it is understood that results can be obtained for an image at times ranging from 0.1307 to 0.3391 seconds for 2-fold cross-validation and from 1.1235 to 2.8945 seconds for 10-fold cross-validation, in direct proportion to the image data sizes.

The results obtained in the experiments using a 23-layer CNN classifier and data augmentation are shown between Tables 9 and 11 for 2-fold cross-validation. When these tables are examined, the highest mean sensitivity, specificity, accuracy, F-1 score, and AUC values are 0.8855, 0.9675, 0.9321, 0.8782, 0.9736; respectively for 5 times increment, 0.8959, 0.9699, 0.9410, 0.8917, 0.9780; respectively for 10 times increment.
and 0.8870, 0.9650, 0.9367, 0.8828, 0.9835; respectively for 20 times increment. It is seen that, in obtaining these highest mean values, the input image combinations of LBP, LE, and Original+LBP+LE for 5 times increment; LBP, Original+GLCM+LE, and Original+GLCM+LBP+LE for 10 times increment and LE, Original+LBP+LE, and Original+GLCM+LBP+LE for 20 times increment were used. When the processing times are examined, it is understood that results can be obtained for an image at times ranging from 0.6438 to 1,6378 seconds for 5 times increment, from 1,2964 to 3,5341 seconds for 10 times increment, and from 2,6017 to 16,3373 seconds for 20 times increment, in direct proportion to the image data sizes.

When Tables 12 and 13, which includes the results obtained by using the Alexnet CNN classifier, are examined within the scope of the study, the highest mean sensitivity, specificity, accuracy, F-1 score, and AUC values were found to be 0.9041, 0.9620, 0.9457, 0.9020, 0.9767; respectively for 2-fold cross-validation, 0.9301, 0.9814, 0.9587, 0.9234, 0.9820; respectively for 10-fold cross-validation. In obtaining these highest average values mentioned, it is seen that the combinations of input images such as LE and Original+GLCM+LBP+LE for 2-fold cross-validation and 10-fold cross-validation. When the processing times are examined, it is understood that results can be obtained for an image at times ranging from 0.0516 to 0.1062 seconds for 2-fold cross-validation and from 0.4003 to 0.8628 seconds for 10-fold cross-validation, in direct proportion to the image data sizes.

When Tables 14 and 15, which includes the results obtained by using the Mobilenetv2 CNN classifier, are examined within the scope of the study, the highest mean sensitivity, specificity, accuracy, F-1 score, and AUC values were found to be 0.9197, 0.9691, 0.9473, 0.9058, 0.9783 respectively for 2-fold cross-validation, 0.9404, 0.9782, 0.9599, 0.9284, 0.9860 respectively, for 10-fold cross-validation. In obtaining these highest average values mentioned, it is seen that the combinations of input images such as Original, LE, and Original+GLCM for 2-fold cross-validation, the Original, GLCM and LE were used for 10-fold cross-validation. When the processing times are examined, it is understood that results can be obtained for an image at times ranging from 0.7366 to 0.7987 seconds for 2-fold cross-validation and from 6.5895 to 7.1508 seconds for 10-fold cross-validation, in direct proportion to the image data sizes.

Table 16 contains the comparison of the results obtained in the literature studies, of which a piece of summary information was given in the introduction section, with the results obtained within the scope of the study. As can be seen from Table 16, the highest mean sensitivity, specificity, accuracy, F-1 score, and AUC values obtained within the scope of

| Study                        | SEN       | SPE       | ACC       | F-1       | AUC       |
|-----------------------------|-----------|-----------|-----------|-----------|-----------|
| Han et al.[20]              | 0.968-0.979 | X         | 0.968-0.979| 0.968-0.979| 0.982-0.990|
| Ardakani et al. [21]        | 0.7843-1.000| 0.6863-1.000| 0.7892-0.9951| X         | 0.894-0.994|
| Jaiswal et al. [22]         | 0.9206-0.9735| 0.8972-0.9621| 0.909-0.9625| 0.9109-0.9629| 0.97      |
| Pathak et al. [23]          | 0.9146    | 0.9478    | 0.9302    | X         | X         |
| Ouyang et al. [24]          | 0.869     | 0.901     | 0.875     | 0.820     | 0.944     |
| Sakagianni et al. [25]      | 0.8831    | X         | X         | 0.8831    | X         |
| Hu et al. [26]              | 0.8330    | 0.9560    | 0.9060    | X         | 0.9430    |
| Our Study (2- fold)         | 0.9197    | 0.9891    | 0.9473    | 0.9058    | 0.9888    |
| Our Study (10- fold)        | 0.9404    | 0.9901    | 0.9599    | 0.9284    | 0.9903    |
the study were 0.9197, 0.9891 (the second highest), 0.9473, 0.9058, 0.9888 respectively, for 2-fold cross-validation and 0.9404, 0.9901 (the second highest), 0.9599, 0.9284, 0.9903 respectively for 10-fold cross-validation.

5 Discussion

Important results were obtained within the scope of this study, which recommends automatic classification of CT images of lungs for early diagnosis of Covid-19 disease and benefits from texture analysis methods, machine learning and deep learning. Within the scope of the study, the classification success of texture analysis methods, machine learning and deep learning methods were compared comprehensively for many method combinations.

Within the scope of the study, if the SVM and k-NN classifiers are used as machine learning methods are compared, it is seen that the SVM classifier is more successful than the k-NN classifier. As for the classification time per image, it is understood that the SVM classifier for 2-fold cross-validation and the k-NN classifier for 10-fold cross-validation produce faster classification results. When the results obtained by using only the original images for SVM and k-NN classifiers are compared with the highest classification results obtained by adding new texture feature images, it is understood that significant improvements were achieved in many comparison parameters. Using new texture feature images, improvements were observed in the sensitivity, specificity, accuracy, F-1 score, and AUC parameters for SVM classifier and 2-fold cross-validation by 0.5181%, 1.2871% (with reference to the second-highest result), 0.8596%, 1.3805%, 0.3817% respectively. These improvements were; 1.4851% (with reference to the second-highest result), 0.5731%, 0.8552%, 0.2096% respectively, in the specificity, accuracy, F-1 score, and AUC parameters for the SVM classifier and 10-fold cross-validation. The highest sensitivity values for 2-fold cross-validation and 10-fold cross-validation using the k-NN classifier were obtained using the original images. However, it is seen that the use of new texture feature images reflected positively on other comparison parameters. Using the new texture feature images, it is seen that improvements in the specificity, accuracy, F-1 score, and AUC parameters for the k-NN classifier and 2-fold cross-validation by 2.3762% (with reference to the second-highest result), 0.7163%, 0.8110%, 0.8871% respectively and 1.9802% (with reference to the second highest result), 0.3582%, 0.1163%, 0.9370% respectively for 10-fold cross-validation were achieved. In this context, it is understood that the direct use of GLCM and LE texture feature extraction methods in the improvement of the classification results using the SVM classifier, and the use of LBP and LE tissue feature extraction methods in combination with the original images in the improvement of the classification results using the k-NN classifier will provide the most effective benefit.

Within the scope of the study, the effect of increasing the number of images used in education by using data augmentation on the classification results for the CNN classifier was examined. Because of the significant increases in training load and classification time due to data augmentation, the experiments were realized only for the use of 23-layer CNN and 2-fold cross-validation. The results of the study show that data augmentation has partial positive contributions on the classification. However, it is understood that it causes the same rates of increases in classification times as well. Also, when the results obtained by using only the original images for 23-layer CNN and data augmentation are compared with the
highest classification results obtained by adding new texture feature images, it is understood that significant improvements were achieved in many comparison parameters.

If a comparison of the CNN architectures used within the scope of the study is required, it is seen that the most successful results are obtained by using Mobilenetv2. In terms of classification times though, it is understood that the fastest results are achieved by using Alexnet and 23-layer CNN. There is a sevenfold difference between Alexnet and Mobilenetv2 in terms of the speed of achieving results. Similar to the cases of SVM and k-NN, it is understood that the inclusion of new texture feature images into the classification has positive effects on the classification results for CNN classifiers, as well. Some conclusions were obtained for CNN classifiers, by comparing the results obtained by using only the original images with the highest classification results obtained by adding new texture feature images. In this context, using new tissue feature images, it was seen that improvements were achieved in the sensitivity, specificity, accuracy, F-1 score, and AUC parameters by 0.4145%, 3.3069%, 1.3467%, 1.8929%, 0.6367% respectively, for 23-layer CNN classifier and 2-fold cross-validation. The highest sensitivity and F-1 score values for 10-fold cross-validation using a 23-layer CNN classifier were obtained using the original images. However, it is seen that the use of new texture feature images reflected positively on other comparison parameters. In this context, it is seen that by using new texture feature images, improvements in the specificity, accuracy, and AUC parameters for a 23-layer CNN classifier and 10-fold cross-validation by 1.7822%, 0.0430%, 0.5768%, respectively, were achieved. For the Alexnet classifier, those improvements were even higher. Using new tissue feature images, improvements in the sensitivity, specificity, accuracy, F-1 score, and AUC parameters for the Alexnet classifier and 2-fold cross-validation by 3.7824%, 3.9010%, 3.8395%, 6.3918%, 2.4969% respectively, and for 10-fold cross-validation, by 2.1762%, 2.5347%, 1.5903%, 2.5606%, and 0.8262% respectively, were achieved. The highest sensitivity value for 2-fold cross-validation using the Mobilenetv2 classifier was obtained using the original images. However, it is seen that the use of new texture feature images reflected positively on other comparison parameters. In this context, using new texture feature images, it is seen that improvements in the specificity, accuracy, F-1 score, and AUC parameters for the Mobilenetv2 classifier and 2-fold cross-validation by 1.3861%, 0.1862%, 0.2644%, 0.2356%, respectively were achieved. The highest sensitivity, accuracy, and F-1 score value for 10-fold cross-validation using the Mobilenetv2 classifier was obtained using the original images. However, it is understood that using the new texture feature images provided improvements in the specificity and AUC parameters for the Mobilenetv2 classifier and 10-fold cross-validation by 1.0891% and 0.2682%, respectively.

If deep learning classifiers are compared with machine learning classifiers, it is seen that deep learning classifiers are more successful than machine learning classifiers. In addition, deep learning classifiers produce results at least 5 times faster than machine learning classifiers.

When the results of the study obtained within the scope of the study and summary information of which is given in Table 16 are compared with the results obtained in previous studies in the literature, it is seen that successful results were obtained. In this context, it is understood that higher results were obtained than the results obtained in the studies performed by Pathak et al. [23], Ouyang et al. [24], Sakagianni et al. [25], and Hu et al. [26]. Study results are generally equivalent to the results obtained in the study by Jaiswal et al. [22] and behind the studies performed by Han et al. [20] and Ardakani et al. [21]. However, for making a fairer comparison, it should be taken into account that in the study conducted...
by Ardakani et al. [21], the determination of $60 \times 60$ ROI areas on the image was done manually and dense areas were marked. Indeed, although similar transfer learning approaches were used by Jaiswal et al. [22], poorer results were obtained because the image was used as a whole. In our study, images were classified as a whole as well. One of the most important implications of this study is that if the images obtained by texture feature methods are used directly or together with the original ones, significant improvements can be achieved in the study results. This situation is applicable to other studies with high results. In this way, it is considered that the results obtained in other studies can be enhanced even more.

The results of the study indicate that analyzing CT images of lungs by the help of deep learning methods in diagnosing Covid-19 disease will speed up the diagnosis and significantly reduce the burden on healthcare workers. In this context, it is critical to increase the number of radiological and clinical data of Covid-19 patients and to make them available to researchers through open access in order to improve these studies and obtain better results.

In the studies to be carried out after this stage, it is aimed to make automatic classification of X-Ray images of chest, which is an important diagnostic tool such as CT images of lung, by the help of methods based on deep learning, in Covid-19 disease diagnosis. In addition, it will be tried to increase the success of study by making use of complex-valued CNN and transfer learning approaches. Finally, it is evaluated that another important study topic is the realization of pipeline methods that will provide better results by combining the results obtained using different extraction property methods.

Compliance with Ethical Standards

Conflict of interests Dr. Ceylan declares that he has no conflict of interest. Mr. Yasar declares that he has no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI (Access Time: 01 September 2020)
2. Albarello F, Pianura E, Di Stefano F, Cristofaro M, Petrone A, Marchioni L, Palazzolo C, Schinina V, Nicostriti E, Petrosillo N, Campioni P, Eskild P, Zumla A, Ippolito G. COVID 19 INMI Study Group (2020) 2019-novel Coronavirus severe adult respiratory distress syndrome in two cases in Italy: An uncommon radiological presentation. Int J Infect Dis 93:192–197. https://doi.org/10.1016/j.ijid.2020.02.043
3. Altman NS (1992) An introduction to kernel and nearest-neighbor nonparametric regression. Am Stat 46(3):175–185. https://doi.org/10.2307/2685209
4. Ardakani AA, Kanafi AR, Acharya UR, Khadem N, Mohammadi A (2020) Application of deep learning technique to manage COVID-19 in routine clinical practice using CT results: Results of 10 convolutional neural networks. Comput Biol Med 103795:121. https://doi.org/10.1016/j.compbiomed.2020.103795
5. Armato III SG, McLennan G, Bidaut L, McNitt-Gray MF, Meyer CR, Reeves AP, Clarke LP (2015) Data From LIDC-IDRI. The Lung Imaging Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): A completed reference database of lung nodules on CT scans. Med Phys 38:915–931. https://doi.org/10.1118/1.3528204
6. Bloice MD, Holzinger A (2019) Patch augmentation: Towards efficient decision boundaries for neural networks. arXiv:1911.07922
8. Bloice MD, Roth PM, Holzinger A (2019) Biomedical image augmentation using Augmentor. Bioinformatics 35(21):4522–4524. https://doi.org/10.1093/bioinformatics/btz259
9. Bloice MD, Stocker C, Holzinger A (2017) Augmentor: An image augmentation library for machine learning. arXiv:1708.04680
10. Chen H, Guo J, Wang C, Luo F, Yu X, Zhang W, Li J, Zhao D, Xu D, Gong Q, Liao J, Yang H, Hou W, Zhang Y (2020) Clinical characteristics and intraterine vertical transmission potential of COVID-19 infection in nine pregnant women: a retrospective review of medical records. The Lancet 395(10226):809–815. https://doi.org/10.1016/S0140-6736(20)30360-3
11. Cheng SC, Chang YC, Chiang YLF, Chien YC, Cheng M, Yang CH, Huang CH, Hsu YN (2020) First case of Coronavirus Disease 2019 (COVID-19) pneumonia in Taiwan. J Formos Med Assoc 119(3):747–751. https://doi.org/10.1016/j.jfma.2020.02.007
12. Clark K, Vendt B, Smith K, Freymann J, Kirby J, Koppel P, Moore S, Phillips S, Maffitt D, Pringle M, Tarbox L, Prior F (2013) The Cancer Imaging Archive (TCIA): Maintaining and operating a public information repository. J Digit Imaging 26(6):1045–1057. https://doi.org/10.1007/s11727-013-9622-7
13. Cohen JP, Morrison P, Dao L (2020) COVID-19 image data collection, arXiv, 2020. https://github.com/ieee8023/covid-chestxray-dataset
14. Han Z, Wei B, Hong Y, Li T, Cong J, Zhu X, Zhang W (2020) Accurate screening of COVID-19 using attention based deep 3D multiple instance learning. IEEE Trans Med Imaging 39(8):2584–2594. https://doi.org/10.1109/TMI.2020.2996256
15. Hardalac F, Yasar H, Akyel A, Kutbay U (2020) A novel comparative study using multi-resolution transforms and convolutional neural network (CNN) for contactless palm print verification and identification. Multimedia Tools and Applications. https://doi.org/10.1007/s11042-020-09005-2
16. Hu X, Chen J, Jiang X, Tao S, Zhen Z, Zhou C, Wang J (2020) CT imaging of two cases of one family cluster 2019 novel coronavirus (2019-nCoV) pneumonia: inconsistency between clinical symptoms amelioration and imaging sign progression. Quant Imaging Med Surg 10(2):508. https://doi.org/10.21037/qims.2020.02.10
17. Hu S, Gao Y, Niu Z, Jiang Y, Li L, Xiao X, Ye H (2020) Weakly supervised deep learning for COVID-19 infection detection and classification from CT images. IEEE Access 8:118869–118883. https://doi.org/10.1109/ACCESS.2020.3005510
18. Huang C, Wang Y, Li X, Ren L, Zhao J, Hu Y, Li Zhang, Fan G, Xu J, Gu X, Cheng Z, Yu T, Xia J, Wei Y, Wu W, Xie X, Yin W, Li H, Liu M, Xiao Y, Gao H, Guo L, Xie J, Wang G, Jiang R, Gao Z, Lin Q, Wang J, Cao B (2020) Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. The Lancet 395(10223):497–506. https://doi.org/10.1016/S0140-6736(20)30183-5
19. Jaiswal A, Gianchandani N, Singh D, Kumar V, Kaur M (2020) Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning. Journal of Biomolecular Structure and Dynamics 1–8. https://doi.org/10.1080/07391102.2020.1788642
20. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: In Advances in neural information processing systems, pp 1097–1105
21. Li W, Cui H, Li K, Fang Y, Li S (2020) Chest computed tomography in children with COVID-19 respiratory infection. Pediatric Radiology. https://doi.org/10.1007/s00247-020-04656-7
22. Li M, Lei P, Zeng B, Li Z, Yu P, Fan B, Wang C, Li Z, Zhou J, Hu S, Liu H (2020) Coronavirus Disease (COVID-19): Spectrum of CT Findings and Temporal Progression of the Disease. Academic Radiology. https://doi.org/10.1016/j.acra.2020.03.003
23. Lim J, Jeon S, Shin HY, Kim MJ, Seong YM, Lee WJ, Choe KW, Kang YM, Lee B, Park SJ (2020) Case of the index patient who caused tertiary transmission of COVID-19 infection in Korea: the application of lopinavir/ritonavir for the treatment of COVID-19 infected pneumonia monitored by quantitative RT-PCR. J Korean Med Sci 35(6):e79. https://doi.org/10.3346/jkms.2020.35.e79
24. Lin C, Ding Y, Xie B, Sun Z, Li X, Chen Z, Niu M (2020) Asymptomatic novel coronavirus pneumonia patient outside Wuhan: The value of CT images in the course of the disease. Clin Imaging 63:7–9. https://doi.org/10.1016/j.clinimag.2020.02.008
25. Liu KC, Xu P, Lv WF, Qiu XH, Yao JL, Jin-Feng G (2020) CT manifestations of coronavirus disease-2019: a retrospective analysis of 73 cases by disease severity. Eur J Radiol 126:108941. https://doi.org/10.1016/j.ejrad.2020.108941
26. Long JB, Ehrenfeld JM (2020) The role of augmented intelligence (ai) in detecting and preventing the spread of novel coronavirus. J Med Syst 44:59. https://doi.org/10.1007/s10916-020-1536-6
27. Ojala T, Pietikäinen M, Harwood D (1996) A comparative study of texture measures with classification based on featured distributions. Pattern Recogn 29(1):51–59. https://doi.org/10.1016/0031-3203(95)00067-4
28. Ornek AH, Ceylan M (2019) Comparison of traditional transformations for data augmentation in deep learning of medical thermography. In: International Conference on Telecommunications and Signal Processing (TSP), pp 191–194. https://doi.org/10.1109/TSP.2019.8769068
29. Ornek AH, Ceylan M, Ervural S (2019) Health status detection of neonates using infrared thermography and deep convolutional neural networks. Infrared Physics & Technology 103:103044. https://doi.org/10.1016/j.infrared.2019.103044
30. Ouyang X, Huo J, Xia L, Shan F, Liu J, Mo Z, Shi F (2020) Dual-sampling attention network for diagnosis of COVID-19 from community acquired pneumonia. IEEE Trans Med Imaging 39(8):2595–2605. https://doi.org/10.1109/TMI.2020.2995508
31. Pan Y, Guan H, Zhou S, Wang Y, Li Q, Zhu T, Hu Q, Xia L (2020) Initial CT findings and temporal changes in patients with the novel coronavirus pneumonia (2019-nCoV): a study of 63 patients in Wuhan, China. European radiology. https://doi.org/10.1007/s00330-020-06731-x
32. Pathak Y, Shukla PK, Tiwari A, Stalin S, Singh S, Shukla PK (2020) Deep Transfer Learning based Classification Model for COVID-19 Disease. IRBM. https://doi.org/10.1016/j.irbm.2020.05.003
33. Qin C, Liu F, Yen TC, Lan X (2020) 18 F-FDG PET/CT findings of COVID-19: a series of four highly suspected cases. European Journal of Nuclear Medicine and Molecular Imaging 47(5):1281-1286. https://doi.org/10.1016/j.ejnm.2020.03.004
34. Sakagianni A, Feretzakis G, Kalles D, Koufopoulou C, Kaldis V (2020) Setting up an easy-to-use machine learning pipeline for medical decision support: Case study for COVID-19 diagnosis based on deep learning with CT scans. Stud Health Technol Inform 272:13–16. https://doi.org/10.3233/SHTI200481
35. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) Mobilenetv2: Inverted residuals and linear bottlenecks, pp 4510–4520. https://doi.org/10.1109/CVPR.2018.00474
36. Singhal T (2020) A review of coronavirus disease-2019 (COVID-19). The Indian J Pediatr 87(4):281–286. https://doi.org/10.1007/s12098-020-03263-6
37. Shen C, Yu N, Cai S, Zhou J, Sheng J, Liu K, Zhou H, Guo Y, Niu G Quantitative computed tomography analysis for stratifying the severity of Coronavirus Disease 2019. Journal of Pharmaceutical Analysis, https://doi.org/10.1016/j.jpha.2020.03.004, vol 2020
38. Vapnik V, Chervonenkis A (1964) A note on one class of perceptrons. Autom Remote Control 25
39. Xia W, Shao J, Guo Y, Peng X, Li Z, Hu D (2020) Clinical and CT features in pediatric patients with COVID-19 infection: Different points from adults. Pediatric Pulmonology. https://doi.org/10.1002/ppul.24718
40. Xing Z, Jia H (2020) An improved thermal exchange optimization based GLCM for multi-level image segmentation. Multimed Tools Appl 79:12007–12040. https://doi.org/10.1007/s11042-019-08566
41. Xu YH, Dong JH, An WM, Lv XY, Yin XP, Zhang JZ, Dong L, Ma X, Zhang HJ, Gao BL (2020) Clinical and computed tomographic imaging features of Novel Coronavirus Pneumonia caused by SARS-CoV-2. J Infect 80(4):394–400. https://doi.org/10.1016/j.jinf.2020.02.017
42. Xu X, Yu C, Qu J, Zhang L, Jiang S, Huang D, Chen B, Zhang Z, Guan W, Ling Z, Jiang R, Hu T, Ding Y, Lin L, Gan Q, Luo L, Tang X, Liu J (2020) Imaging and clinical features of patients with 2019 novel coronavirus SARS-CoV-2. Eur J Nucl Med Mol Imaging 47(5):1275–1280. https://doi.org/10.1007/s00259-020-04735-9
43. Yang W, Cai L, Wu F (2020) Image segmentation based on gray level and local relative entropy two dimensional histogram. Plos one 15(3):e0229651. https://doi.org/10.1371/journal.pone.0229651
44. Yasar H, Ceylan M (2016) A novel approach for reduction of breast tissue density effects on normal and abnormal masses classification. J Med Imaging Health Infor 6(3):710–717. https://doi.org/10.1166/jmihi.2016.1737
45. Zhao J, Zhang Y, He X, Xie P (2020) COVID-CT-Dataset: a CT scan dataset about COVID-19. arXiv:2003.13865
46. Zhu N, Zhang D, Wang W, Li X, Yang B, Song J, Zhao X, Huang B, Shi W, Lu R, Niu P, Zhan F, Ma X, Wang D, Xu W, Wu G, Gao GF, Tan W (2020) A novel coronavirus from patients with pneumonia in China, 2019. New England J Med 382:727–733. https://doi.org/10.1056/NEJMoa2001017

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.
Huseyin Yasar received his B.Sc. and M.Sc. degrees in electrical and electronics engineering from the Selcuk University, Konya, Turkey, in 2010, and 2015, respectively. He has been working electrical and electronics engineering at the Ministry of Health of Republic of Turkey since 2012. His current research interests include image processing, medical image processing, biomedical engineering, and artificial intelligence.

Murat Ceylan received B.Sc. degree in 2000, M.Sc. degree in 2004 and Ph.D. degree in 2009 from Electrical–Electronics Engineering Department of Selçuk University, Konya, Turkey. His current research interests are digital image processing, biomedical systems, artificial intelligence, deep learning, transfer learning and thermal imaging and hyperspectral imaging for biomedical applications.

Affiliations

Huseyin Yasar¹ · Murat Ceylan²

Murat Ceylan
mceylan@ktun.edu.tr

¹ Ministry of Health of Republic of Turkey, Ankara, Turkey
² Faculty of Engineering and Natural Sciences, Department of Electrical and Electronics Engineering, Konya Technical University, Konya, Turkey