



COVID-19 discrimination framework for X-ray images by considering radiomics, selective information, feature ranking, and a novel hybrid classifier[☆]

Hasan Koyuncu^{*}, Mücahid Barstuğan

Konya Technical University, Faculty of Engineering and Natural Sciences, Electrical & Electronics Engineering Department, Konya, Turkey

ARTICLE INFO

Keywords:

Binary categorization
Chaotic
Coronavirus
Framework design
Hybrid classifier
Optimization

ABSTRACT

In medical imaging procedures for the detection of coronavirus, apart from medical tests, approval of diagnosis has special significance. Imaging procedures are also useful for detecting the damage caused by COVID-19. Chest X-ray imaging is frequently used to diagnose COVID-19 and different pneumonias. This paper presents a task-specific framework to detect coronavirus in X-ray images. Binary classification of three different labels (healthy, bacterial pneumonia, and COVID-19) was performed on two differentiated data sets in which corona is stated as positive. First-order statistics, gray level co-occurrence matrix, gray level run length matrix, and gray level size zone matrix were analyzed to form fifteen sub-data sets and to ascertain the necessary radiomics. Two normalization methods are compared to make the data meaningful. Furthermore, five feature ranking approaches (*Bhattacharyya*, *entropy*, *Roc*, *t-test*, and *Wilcoxon*) are mentioned to provide necessary information to a state-of-the-art classifier based on Gauss-map-based chaotic particle swarm optimization and neural networks. The proposed framework was designed according to the analyses about radiomics, normalization approaches, and filter-based feature ranking methods. In experiments, seven metrics were evaluated to objectively determine the results: accuracy, area under the receiver operating characteristic (ROC) curve, sensitivity, specificity, g-mean, precision, and f-measure. The proposed framework showed promising scores on two X-ray-based data sets, especially with the accuracy and area under the ROC curve rates exceeding 99% for the classification of coronavirus vs. others.

1. Introduction

COVID-19 has become a mortal pandemic, and its detection is vital for not only COVID-19 positive individuals but also for people who have been in close proximity to such individuals [1]. The early detection of coronavirus is important to provide necessary medical intervention despite the unknown prognosis [2]. In this context, literature studies on computer-aided diagnosis systems can help medical experts for supporting the diagnosis to be decided [3–5].

Here, we propose a task-specific framework for detecting coronavirus in X-ray images. For this purpose, framework analyses were performed by using four radiomics, two normalization approaches, five filter-based feature selection methods and an efficient optimized classifier. The optimized classifier is kept constant at the end of the framework. These analyses were examined in detail to design the remaining parts of the framework and to realize the highest classification performance that the framework was capable of.

1.1. Review of state-of-the-art methods

Ozturk et al. [6] proposed a COVID-19 detection model based on chest X-ray images, and they performed binary (COVID-19 and no-finding) and multiclass (COVID-19 and no-finding and pneumonia) classifications on two challenging data sets. The first data set included 125 COVID-19 and 500 no-finding images, while the second one involved 125 COVID-19, 500 no-finding, and 500 pneumonia cases. Both data sets were discriminated by DarkCovidNet, which yielded classification accuracies of 98.08% and 87.02% for binary and multiclass classifications, respectively. Toğaçar et al. [7] used a support vector machine (SVM)-based model to classify an X-ray data set including 295 coronavirus, 65 normal, and 98 pneumonia cases. A preprocessing step in which the data classes were restructured using a fuzzy color algorithm was used for the images. The restructured images were then stacked with the original images, and deep learning models were considered to extract features from the data set. Efficient features were

[☆] Acknowledgment: This work is supported by the Coordinatorship of Konya Technical University's Scientific Research Projects.

^{*} Correspondence to: Konya Technical University, Faculty of Engineering and Natural Sciences, Electrical & Electronics Engineering Department, 42250, Konya, Turkey.

E-mail addresses: hkoyuncu@ktun.edu.tr (H. Koyuncu), mbarstugan@ktun.edu.tr (M. Barstuğan).

selected with social mimic optimization (SMO) and classified by the SVM-based model. The proposed model achieved 99.27% accuracy for multiclass classification. Ucar and Korkmaz [8] designed a diagnosis model employing Deep Bayes-SqueezeNet to categorize coronavirus in X-ray images. The data set used by them contained X-ray images belonging to three categories (1583 normal, 4290 pneumonia and 76 COVID-19), and the data were augmented to obtain higher performance. After augmentation, the proposed model achieved an overall accuracy of 98.26% for multiclass classification. Apostolopoulos and Mpesiana [9] compared deep learning methods by considering two X-ray-based data sets: one consisting of 224 COVID-19, 700 bacterial pneumonia, and 504 normal images and the other comprising 224 COVID-19, 714 bacterial and viral, and 504 normal cases. Pretrained models such as VGG19, MobileNet-v2, and Inception-ResNet-v2 were implemented for binary and multiclass classification, and the best accuracy rate (96.78%) was recorded for binary classification with MobileNet-v2. Butt et al. [10] constructed a CT image data set comprising 184 COVID-19, 194 influenza-A viral pneumonia, and 145 healthy cases. They proposed a model employing a three-dimensional (3D) convolutional neural network (CNN) and used it to segment and identify the candidate region. The model achieved an area under the ROC (which stands for “receiver operating characteristic”) curve (AUC) score of 99.6% for coronavirus vs. noncoronavirus classification. Li et al. [11] used 4356 CT images to train a 3D deep learning model named COVNet. The collected data set was divided into two classes: COVID-19 and others. In experiments, COVNet achieved an AUC score of 96% for the detection of coronavirus. Kang et al. [12] used 2522 CT images containing 1495 COVID-19 and 1027 pneumonia cases, and employed the V-Net model was handled to obtain the lung, lung lobes, and pulmonary segments. The lesion region was subsequently segmented. Radiomic and handcrafted features were extracted from each CT image, and five classifiers were used to categorize them. The best classification accuracy (95.5%) was achieved by a neural network (NN) algorithm. Afshar et al. [13] used a model processing capsule network (CN) to classify X-ray images belonging to four types of classes (individuals with COVID-19, those with a viral disease, those with a bacterial disease, and normal individuals). The study focused on coronavirus vs. noncoronavirus classification to detect COVID-19. The CN-based structure achieved 95.7% accuracy. Subsequently, the application of a transfer learning method to the CN increased the accuracy up to 98.3%. Mahdy et al. [14] performed binary classification on X-ray images comprising 15 normal and 25 COVID-19 cases. The study first performed multilevel thresholding on images, and the thresholding results were classified into two categories (COVID-19 and normal) with an SVM. Experiments showed that the suggested pipeline achieved 97.48% accuracy for binary classification. Hemdan et al. [15] used different deep learning methods to classify 25 COVID-19 and 25 normal cases in an X-ray data set. The best classification accuracy (90%) was obtained by the VGG19 and DenseNet201 algorithms.

Table 1 presents a review of state-of-the-art methods.

In addition to diagnoses with imaging modalities, blood tests [16, 17] and clinical data [18] have also been considered for COVID-19 diagnosis.

Deep-learning-based approaches have been frequently used for COVID-19 classification. However, the necessity of a comprehensive-task affirmative framework keeps its currency to achieve remarkable results on coronavirus classification. In particular, there is a need for a framework design for distinguishing samples involving coronavirus from other labels.

Studies on coronavirus detection generally perform binary and multiclass classification. In particular, the binary categorization of multiclass data is important. In other words, the design of a task-specific framework should focus on detection of the target illness, and it should be capable of revealing the sample corresponding to the illness from among other labels, including normal samples. An illness-based classification model based on binary classification of multiclass data pertaining to the illness and other labels can be designed. Furthermore, a

coronavirus-specific framework that is intended solely for the detection of coronavirus can be designed. Herein, this process arises as the second necessity of the literature.

1.2. Framework design review

As the first step in the development of a task-affirmative framework, the radiomic features to be extracted should be identified; these features help achieve reliable performance. At this point, the combination of these features and their in-depth analysis constitute the selective information to be processed among the model. First-order statistics (FOS), gray level co-occurrence matrix (GLCM), gray level run length matrix (GLRLM), and gray level size zone matrix (GLSZM) are efficient radiomics frequently used in tasks such as adrenal tumor classification, esophageal cancer categorization, and brain tumor grading [19–24]. These radiomics have also been used in various imaging modalities such as X-ray, magnetic resonance (MR), computed tomography (CT) imaging [19–24]. Yang et al. [19] designed a pipeline for esophageal cancer classification in X-ray images. GLCM features were evaluated as textural information and seen as one class of feature extraction tools. Liao et al. [20] mentioned that different radiomics obtained in MR images, namely, FOS, GLCM, and GLSZM, apart from others could be used for making survival predictions of patients. Koyuncu et al. [21] used discrete wavelet transform (DWT) and GLCM features to categorize adrenal tumors as binary in 114 CT images involving eight types of tumors. Barstuğan et al. [22] added GLRLM features in addition to DWT and GLCM radiomics to classify adrenal tumors in 122 MR images involving nine types of tumors. Yang et al. [23] graded rheumatoid arthritis in ultrasound images by using a problem-specific model that was developed on the basis of tests with GLCM features and SVM derivatives.

As the second step in the development of a task-oriented model, feature ranking approaches, which are as significant as radiomics, should be chosen since the essential information should be provided to the classifier for achieving the best performance. Raghavendra et al. [24] compared *Bhattacharyya*, *entropy*, *Roc*, *t-test*, and *Wilcoxon* rankings to find the coherent ranking with the classifier unit. In experiments, echocardiographic images were categorized into two classes: normal and ischemic heart disease. Nascimento et al. [25] proposed a special pipeline to classify lymphoma images, and the *Wilcoxon* ranking was used in the feature selection part near of other methods for improving the classification performance. Momenzadeh et al. [26] developed a hidden-Markov-model-based framework by considering *Bhattacharyya*, *entropy*, *Roc*, *t-test*, and *Wilcoxon* rankings to classify microarray data sets. They considered B-cell lymphoma, leukemia cancer, and prostate data sets to evaluate the performance of the framework. Koyuncu et al. [27] considered FOS features along with five feature ranking approaches (*Bhattacharyya*, *entropy*, *Roc*, *t-test*, and *Wilcoxon*) to classify high and low grade gliomas (brain tumors) defined in different MRI-phase combinations.

As the third step in the development of an efficient framework, the classifier unit, which is the most important section, should be evaluated using a state-of-the-art classifier. Koyuncu [28] examined chaotic particle swarm optimization (CPSO) and compared different chaotic maps to determine the most effective CPSO. Gauss-map-based CPSO (GM-CPSO) achieved remarkable convergence against state-of-the-art methods for global function optimization. To test the efficiency of GM-CPSO, Koyuncu hybridized it with a NN, and designing a hybrid classifier was considered as the second task. GM-CPSO–NN was compared with recent and efficient hybrid classifiers for epileptic seizure recognition. In the data set, five categories (tumor area, nontumor area, closed-eye, and open-eye recordings with no seizure activity, and seizure activity) were considered, and the seizure vs. nonseizure discrimination was performed with the highest accuracy (97.24%) by GM-CPSO–NN. In another study of Koyuncu [29], GM-CPSO–NN was compared with robust hybrid NNs and with the derivatives of

Table 1
Review of state-of-the-art method review.

Study	Database	Year	Features extracted	Classification method	Network type
Ozturk et al. [6]	GitHub - Covid	2020	Deep features	Binary classification	CNN
Ozturk et al. [6]	Chest X-ray Data set	2020	Deep features	Multiclass Classification	CNN
Toğaçar et al. [7]	1. GitHub - Covid Chest X-ray Data set 2. Kaggle - COVID-19 Chest X-ray Database	2020	Deep features	Multiclass classification	SVM
Ucar and Korkmaz [8]	1. Kaggle - COVID-19 Chest X-ray Database 2. Kaggle - RSNA Pneumonia Detection Challenge 3. GitHub - Actualmed-COVID-Chest X-ray Data set 4. GitHub - Covid Chest X-ray Data set	2020	Deep features	Multiclass classification	Deep Bayes-SqueezeNet
Apostolopoulos and Mpesiana [9]	Kaggle – X-rays and	2020	Deep features	Binary classification	Pretrained models
Apostolopoulos and Mpesiana [9]	CT Snapshots of	2020		Binary classification	Pretrained models
Apostolopoulos and Mpesiana [9]	COVID-19 Patients	2020		Multiclass classification	Pretrained models
Apostolopoulos and Mpesiana [9]		2020		Multiclass classification	Pretrained models
Butt et al. [10]	Not Declared	2020	Deep features	Multiclass classification	3D CNN
Li et al. [11]	GitHub - COVNet	2020	Deep features	Binary classification	3D CNN
Kang et al. [12]	Chinese Center for Disease control and prevention	2020	Radiomic and Handcrafted features	Binary classification	V-Net
Afshar et al. [13]	1. Kaggle - COVID-19 Chest X-ray Database 2. Kaggle - RSNA Pneumonia Detection Challenge 3. GitHub - Actualmed-COVID-Chest X-ray data set 4. GitHub - Covid Chest X-ray data set	2020	Deep features	Binary classification	Capsule network
Mahdy et al. [14]	1. Montgomery County X-ray data set 2. GitHub - Covid Chest X-ray data set	2020	No feature extraction	Binary classification	SVM
Hemdan et al. [15]	GitHub - Covid Chest X-ray Data set	2020	Deep features	Binary classification	Pretrained models

SVM for Parkinson's disease (PD) classification. Experiments showed that GM-CPSO-NN outperformed all classifiers in seven metric-based tests conducted on two PD data sets. In yet another study, Koyuncu et al. [27] used GM-CPSO-NN as the classifier unit to design a task-affirmative framework for brain tumor classification. Two optimized classifiers were compared with GM-CPSO-NN to identify the classifier that was appropriate for the construction of the framework. According to the results, the framework containing GM-CPSO-NN had the highest accuracy score (90.18%) for the binary characterization of brain tumors.

In this study, four efficient radiomics (FOS, GLCM, GLRLM, and GLSZM) were evaluated to determine the best data combination, to rise the selective information, and to design a task-affirmative framework for binary (coronavirus vs. noncoronavirus) classification. Furthermore, two normalization methods (*minmax* and *z-score*) were compared among radiomics to make the data meaningful. Additionally, five feature ranking approaches (*Bhattacharyya*, *entropy*, *Roc*, *t-test*, and *Wilcoxon*) were used to obtain the necessary information that was to be provided to the classifier unit for achieving the highest performance. GM-CPSO-NN was chosen as the classifier owing to its efficiency in different classification tasks and since it was a state-of-the-art classifier. The salient features of this paper are as follows:

- It presents a comprehensive study on the design of a specific COVID-19 framework by using radiomics, selective information, feature ranking, and an efficient classifier.
- It presents a detailed study on coronavirus vs. noncoronavirus discrimination.
- It is the first study to develop an optimized NN-based framework for COVID-19 categorization.

The paper is organized as follows. The GM-CPSO-NN method, radiomics and feature ranking approaches used in this study, and general scheme of framework analysis are presented in Section 2. Information on the data set used in this study, the experimental analysis performed, and interpretations are discussed in Section 3. Concerning the experimental results, proposed framework is taken form and presented in Section 4 beside of the literature comparison. Finally, Section 5 provides the concluding remark.

2. Methods

In this section, the design of the optimized classifier (GM-CPSO-NN) is described, the radiomics and feature ranking approaches used are briefly discussed, and the overall flow and specifications of the designed framework are visually interpreted.

2.1. Gauss-map-based chaotic particle swarm optimization – Neural network

GM-CPSO was developed by Koyuncu [28] through chaotic behavior analysis performed with a particle swarm optimization (PSO) algorithm. GM-CPSO uses (1) and (2) to obtain the new velocity and new position vectors, respectively [28].

$$V_i(t+1) = \omega V_i(t) + c_1 r_1(t) (X_{pbest(i)}(t) - X_i(t)) + c_2 r_2(t) (X_{gbest}(t) - X_i(t)) \quad (1)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (2)$$

In (1) and (2), $V_i(t)$ and $V_i(t+1)$ are the current and new velocities, while $X_i(t)$ and $X_i(t+1)$ denote the current and new positions, respectively. Here, the velocity can be defined as the step size that limits

Table 2

Pseudocode of GM-CPSO.

```

- Determine the parameter values and ranges
- Assign the first positions
- Calculate the error or profit
- Define  $X_{pbest(i)}(0) = X_i(0)$  and  $X_{gbest}(0) = \text{best}(X_{pbest(i)}(0))$ 
- Update the inertia weight vector according to the maximum iteration number  $\rightarrow$  (3)
While (iteration < maximum iteration number)
- Computation of new velocity and position vectors  $\rightarrow$  (1) and (2)
- Calculate the error or profit
  If (minimum error or maximum profit is reached)
    - Save the best particle vector and its fitness
    - Break
  End
- Attain the  $X_{pbest}$  vector for every particle and  $X_{gbest}$  for the entire swarm
- iteration++
End

```

the movement of position. $X_{pbest(i)}(t)$ and $X_{gbest}(t)$ are the individual best position of the i th particle and the global best solution (position) for the entire swarm (all particles), respectively, c_1 and c_2 are the acceleration constants moving the particles to the individual and global best solutions, r_1 and r_2 are random numbers generated within the range (0,1), and ω is the inertia weight restricting the velocity acts. In PSO, ω is frequently set to be a constant value or a variable that is iteratively changed according to the linear descent movement. In GM-CPSO, ω is a variable and it is adjusted according to the Gauss map defined in (3) below. Here, the initial value ($\omega(0)$) is set to 0.7 to achieve robust chaotic behavior [28–30].

$$\omega(i+1) = \begin{cases} 1 & \omega(i) = 0 \\ \frac{1}{\text{mod}(\omega(i), 1)} & \text{otherwise} \end{cases}, \quad \omega(0) = 0.7 \quad (3)$$

As evident in (3), the Gauss map can be equal to zero during iterations, which eliminates the effect of the current velocity on the new velocity. Thus, solution diversity is upgraded by using or eliminating the current velocity phenomenon, which directly effects the new velocity and indirectly the new position. Table 2 presents the pseudocode of GM-CPSO.

An NN generally uses backpropagation-type feedback methods. An error metric, the mean squared error (MSE) being usually preferred, is used to detect the fitness of networks according to the desired and obtained outputs. Backpropagation-type methods can cause fluctuations in the error, and optimization algorithms are employed to improve the convergence of the error rate toward the minimum level [31]. In optimized NNs, the optimization algorithm is generally changed using the feedback algorithm of the NN. The formed structure is defined as hybrid, hybridized, or optimized NN. To simplify the presentation of optimized NNs, we can focus on the flow of the optimization algorithm.

In every iteration of the training phase, the optimized NN evaluates different networks, and the network number is specified as the particle number of the optimization algorithm. Different networks are compared to improve the classification of the patterns at each iteration. Furthermore, these networks are generated according to the phenomena declared in optimization algorithm.

In optimized NNs, every network includes different weight-bias values, and these networks are generally transformed into vectors to be processed by optimization algorithm. In other words, every network is represented as a weight-bias vector. In particular, a network of the optimized NN can be referred to as a particle, position, or weight-bias vector.

Briefly, in the hybrid method, the *calculation of the fitness* in the optimization approach (e.g., the third and seventh items in Table 2) involves the operation of NNs and the determination of MSE rates for each network used [27,28].

In optimized NNs, the flowchart of the optimization algorithm can be considered to describe the training phase in which the aim is to

detect the best weight-bias vector (network). The test part consists of only the evaluation of test data with the best network determined by the optimization algorithm. In other words, the test part employs the best weight-bias vector (the best network) yielding the minimum MSE found in the training phase, to test the system performance. Notably, the performance of optimized NNs depends on the quality of the optimization approach used.

In addition to the parameters of the optimization algorithm, the optimized NNs involve the general NN parameters (hidden node number, maximum iteration number, and minimum error). Concerning the general structure of optimized NNs, GM-CPSO-NN algorithm can be revealed as defined in Fig. 1 by originating the flow of GM-CPSO in training part of the classifier. As declared before, every network (weight-bias vector) can be considered as a particle (position vector) in the optimization method [27–29,31].

In Fig. 1, *parameter definitions* include the aforementioned PSO parameters besides NN parameters. [X_{min} , X_{max}] are defined as the boundaries of weight-bias vectors, and it is advisable to choose the velocity boundaries ($[V_{min}$, $V_{max}]$) as 0.2 times the position boundaries for better convergence [31]. Furthermore, the total network number is defined to be equal to the population size (particle number) of the optimization algorithm. For the achievement of the best classification performance, the population size, hidden node number, acceleration constants, and maximum iteration number should be analyzed in detail.

2.2. Texture radiomics

FOS are the *mean, standard deviation, skewness, kurtosis, energy, and entropy*, obtained from a histogram or acquired directly from an image. As one of the most popular and preferred radiomics, FOS features do not vary among studies [27]. However, the GLCM, GLRLM, and GLSZM features can differ from one study to another, depending on work preferences or the advice of the literature. Concerning this, we need to explain the GLCM, GLRLM, and GLSZM features. For details on the determination of FOS features, the reader is referred to ref. [27].

Information acquired on a single pixel value does not provide detailed information about the texture of an image. Although statistical measurements are easy to obtain, there is no distinctive information obtained via single-pixel-based analysis. The GLCM [32] is based on the pixel intensity distribution of an image, and is part of second-order statistics. The co-occurrence matrix, which provides information on pixel neighborhoods, contains significant information about the positions of pixels with similar gray levels [32]. It is defined as $P = [p(i,j) \ d, \theta]$ and is used to determine the frequency features of pixels (gray value i) and their neighboring pixels (gray value j) at distance d and in direction θ [32]. In this study, 19 efficient features were extracted using the GLCM method; *angular secondary moment, contrast, correlation, autocorrelation, entropy, cluster prominence, cluster shade, dissimilarity, sum of squares: variance, difference variance, inverse difference moment, inverse difference, maximum probability, sum average, sum entropy, sum variance, difference entropy, information measures of correlation 1*, and *information measures of correlation 2* were evaluated in X-ray images. Let N_g be the number of different gray levels in a quantized image, and let $p(i,j)$ denote the (i,j) th input in a normalized gray scale spatial dependence matrix. Let $p_x(i)$ and $p_y(j)$ signify the i th and j th entries in the marginal probability matrix, obtained by summing the rows and columns of $p(i,j)$. Furthermore, let μ_x and μ_y be the mean values of p_x and p_y , and let σ_x and σ_y represent their standard deviations, respectively. The features used can then be expressed as follows [32–34].

$$p_y(j) = \sum_{i=1}^{N_g} p(i,j) \quad (4)$$

$$p_{x+y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j), \quad i+j=k, \quad k=2,3,\dots,2N_g \quad (5)$$

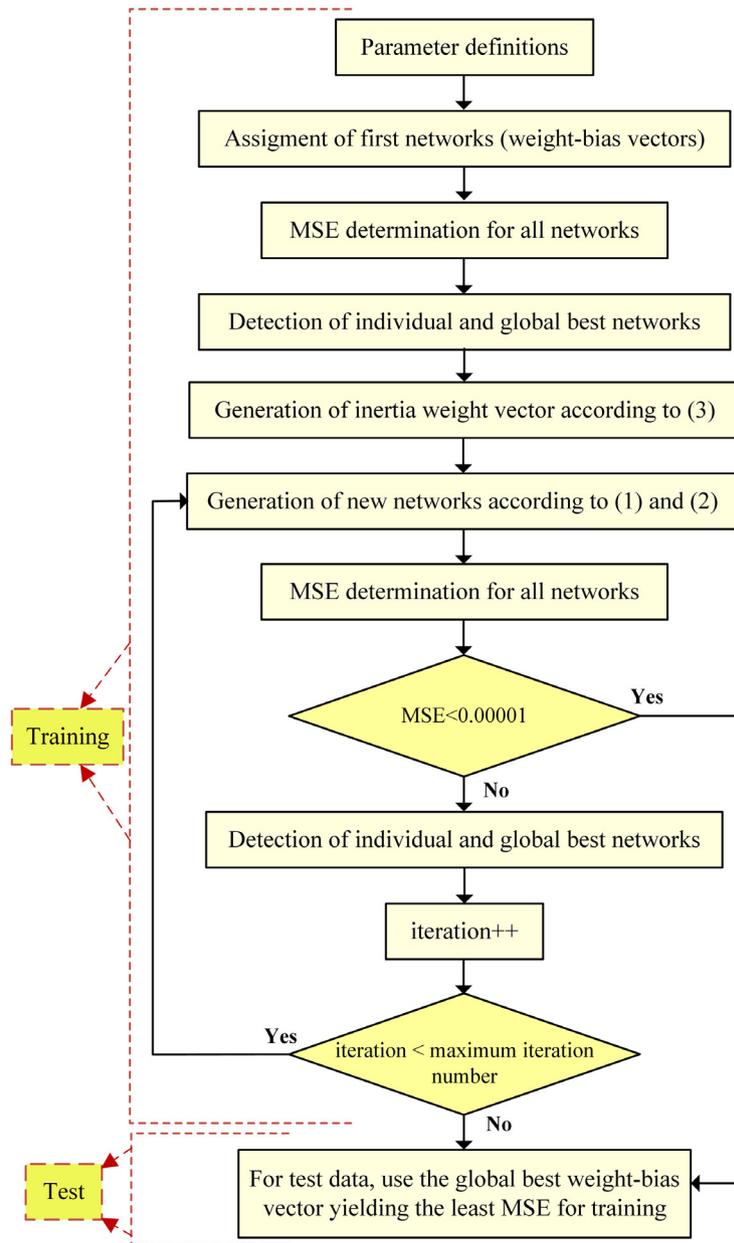


Fig. 1. Flowchart of GM-CPSO-NN.

$$p_{x-y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j), \quad |i - j| = k, \quad k = 0, 1, \dots, N_g - 1 \quad (6)$$

$$HXY = - \sum_i \sum_j p(i, j) \log(p(i, j)) \quad (7)$$

$$HXY1 = - \sum_i \sum_j p(i, j) \log \{ p_x(i) p_y(j) \} \quad (8)$$

$$HXY2 = - \sum_i \sum_j p_x(i) p_y(j) \log \{ p_x(i) p_y(j) \} \quad (9)$$

$$\text{Angular secondary moment} = \sum_i \sum_j \{ p(i, j) \}^2 \quad (10)$$

$$\text{Contrast} = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \right\}, \quad |i - j| = n \quad (11)$$

$$\text{Correlation} = \frac{\sum_i \sum_j (ij) p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (12)$$

$$\text{Autocorrelation} = \sum_i \sum_j \frac{(p_x - \mu_x)(p_y - \mu_y)}{\sigma_x \sigma_y} \quad (13)$$

$$\text{Entropy} = - \sum_i \sum_j p(i, j) \log(p(i, j)) \quad (14)$$

$$\text{Cluster prominence} = \sum_i \sum_j (i + j - \mu_x - \mu_y)^4 p(i, j) \quad (15)$$

$$\text{Cluster shade} = \sum_i \sum_j (i + j - \mu_x - \mu_y)^3 p(i, j) \quad (16)$$

$$\text{Dissimilarity} = \sum_i \sum_j p(i, j) |i - j| \quad (17)$$

$$\text{Sum of squares: variance} = \sum_i \sum_j (i - \mu)^2 p(i, j) \quad (18)$$

$$\text{Difference variance} = \sum_i i^2 p_{x-y}(i) \quad (19)$$

$$\text{Inverse difference moment} = \sum_i \sum_j \frac{p(i, j)}{1 + (i - j)^2} \quad (20)$$

$$\text{Inverse difference} = \sum_i \sum_j \frac{p(i, j)}{1 + |i - j|} \quad (21)$$

$$\text{Maximum probability} = \text{MAX}_{i,j} p(i, j) \quad (22)$$

$$\text{Sum average} = \sum_{i=2}^{2N_g} i p_{x+y}(i) \quad (23)$$

$$\text{Sum entropy} = - \sum_{i=2}^{2N_g} p_{x+y}(i) \log \{p_{x+y}(i)\} \quad (24)$$

$$\text{Sum variance} = \sum_{i=2}^{2N_g} (i - \text{sum entropy})^2 p_{x+y}(i) \quad (25)$$

$$\text{Difference entropy} = - \sum_{i=0}^{N_g-1} p_{x-y}(i) \log \{p_{x-y}(i)\} \quad (26)$$

$$\text{Information measures of correlation 1} = \frac{HXY - HXY1}{\max \{HX, HY\}} \quad (27)$$

$$\text{Information measures of correlation 2} = (1 - \exp[-2(HXY2 - HXY)])^{1/2} \quad (28)$$

The GLRLM [35] reveals the size of homogeneous runs for each gray level. Let G be the number of gray levels, R be the longest run, and W be the number of pixels. The obtained GLRLM has a size $G \times R$, and the elements of each $p(i, j|\theta)$ give the number of occurrences with direction θ , gray level i , and run length j . This study extracted *short run emphasis* (SRE), *long run emphasis* (LRE), *gray level nonuniformity* (GLN), *run length nonuniformity* (RLN), *run percentage* (RP), *low gray level run emphasis* (LGRE), and *high gray level run emphasis* (HGRE) from the GLRLM [35]. The features used are presented in (29)–(35) [35].

$$SRE = \frac{\sum_{i=1}^G \sum_{j=1}^R \frac{p(i, j|\theta)}{j^2}}{\sum_{i=1}^G \sum_{j=1}^R p(i, j|\theta)} \quad (29)$$

$$LRE = \frac{\sum_{i=1}^G \sum_{j=1}^R j^2 p(i, j|\theta)}{\sum_{i=1}^G \sum_{j=1}^R p(i, j|\theta)} \quad (30)$$

$$GLN = \frac{\sum_{i=1}^G \left(\sum_{j=1}^R p(i, j|\theta) \right)^2}{\sum_{i=1}^G \sum_{j=1}^R p(i, j|\theta)} \quad (31)$$

$$RLN = \frac{\sum_{j=1}^R \left(\sum_{i=1}^G p(i, j|\theta) \right)^2}{\sum_{i=1}^G \sum_{j=1}^R p(i, j|\theta)} \quad (32)$$

$$RP = \frac{1}{N} \sum_{i=1}^G \sum_{j=1}^R p(i, j|\theta) \quad (33)$$

$$LGRE = \frac{\sum_{i=1}^G \sum_{j=1}^R \frac{p(i, j|\theta)}{i^2}}{\sum_{i=1}^G \sum_{j=1}^R p(i, j|\theta)} \quad (34)$$

$$HGRE = \frac{\sum_{i=1}^G \sum_{j=1}^R i^2 p(i, j|\theta)}{\sum_{i=1}^G \sum_{j=1}^R p(i, j|\theta)} \quad (35)$$

The GLSZM [36] is an advanced statistical matrix and used for texture characterization. In the GLSZM, run length principle is considered as the calculation method. The width of the matrix is dependent on the zone sizes/surfaces. If the texture homogeneity increases, the zones increase in size and the matrix width increases. Therefore, the result matrix becomes wider. When the texture contains considerable noise, so single pixel zones, the number of zones is large, sometimes with close gray level values, which appear in a part of the matrix. By contrast, when the texture is homogeneous, as there are less zones, the matrix contains more zeros and is flattened. The GLSZM computes one matrix for all directions, unlike the GLCM and GLRLM approaches. In this study, 12 efficient features were considered for the GLSZM: *small zone emphasis* (SZE), *large zone emphasis* (LZE), *gray level nonuniformity* (GLNz), *zone size nonuniformity* (ZSN), *zone percentage* (ZP), *low gray level zone emphasis* (LGZE), *high gray level zone emphasis* (HGZE), *small*

zone low gray level emphasis (SZLGE), *small zone high gray level emphasis* (SZHGE), *large zone low gray level emphasis* (LZLGE), *large zone high gray level emphasis* (LZHGE), and *gray level variance* (GLV). Let P be a size zone matrix, p be the normalized size zone matrix, N_g be the number of discrete intensity values in the image, N_z be the size of the largest homogeneous region in the volume of interest (VOI), and N_α be the number of homogeneous zones in the images [36]. The evaluated features can then be given by (36)–(49) [36].

$$\mu_i = \sum_{i=1}^{N_g} i \sum_{j=1}^{N_z} p(i, j) \quad (36)$$

$$p(i, j) = P(i, j) / \sum_{i=1}^{N_g} \sum_{j=1}^{N_z} P(i, j) \quad (37)$$

$$SZE = \sum_{i=1}^{N_g} \sum_{j=1}^{N_z} \frac{p(i, j)}{j^2} \quad (38)$$

$$LZE = \sum_{i=1}^{N_g} \sum_{j=1}^{N_z} j^2 p(i, j) \quad (39)$$

$$GLNz = \sum_{i=1}^{N_g} \left(\sum_{j=1}^{N_z} p(i, j) \right)^2 \quad (40)$$

$$ZSN = \sum_{j=1}^{N_z} \left(\sum_{i=1}^{N_g} p(i, j) \right)^2 \quad (41)$$

$$ZP = \sum_{i=1}^{N_g} \sum_{j=1}^{N_z} \frac{p(i, j)}{N_\alpha} \quad (42)$$

$$LGZE = \sum_{i=1}^{N_g} \sum_{j=1}^{N_z} \frac{p(i, j)}{i^2} \quad (43)$$

$$HGZE = \sum_{i=1}^{N_g} \sum_{j=1}^{N_z} i^2 p(i, j) \quad (44)$$

$$SZLGE = \sum_{i=1}^{N_g} \sum_{j=1}^{N_z} \frac{p(i, j)}{i^2 j^2} \quad (45)$$

$$SZHGE = \sum_{i=1}^{N_g} \sum_{j=1}^{N_z} \frac{p(i, j) i^2}{j^2} \quad (46)$$

$$LZLGE = \sum_{i=1}^{N_g} \sum_{j=1}^{N_z} \frac{p(i, j) j^2}{i^2} \quad (47)$$

$$LZHGE = \sum_{i=1}^{N_g} \sum_{j=1}^{N_z} p(i, j) i^2 j^2 \quad (48)$$

$$GLV = \frac{1}{N_g \times N_z} \sum_{i=1}^{N_g} \sum_{j=1}^{N_z} (ip(i, j) - \mu_i)^2 \quad (49)$$

2.3. Feature ranking approaches

This section presents a short summary of five feature ranking approaches (*Bhattacharyya*, *entropy*, *Roc*, *t-test*, and *Wilcoxon*) that are frequently applied to upgrade system performance [27].

The *Bhattacharyya distance* (BD) measures the similarity between two probability distributions, and it is calculated as shown in (50), for two classes with normal distributions. Here, μ_1 and μ_2 are defined as the means of the first and second classes, respectively, while σ_1 and σ_2 are the standard deviations of the classes [26,27].

$$BD = \frac{1}{4} \ln \left[\frac{1}{4} \left(\frac{\sigma_1^2}{\sigma_2^2} + \frac{\sigma_2^2}{\sigma_1^2} + 2 \right) \right] + \frac{1}{4} \left[\frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \right] \quad (50)$$

The *entropy* test (e) is also referred to as Kullback–Leibler distance or divergence, and it is used for classes with normal distributions. The

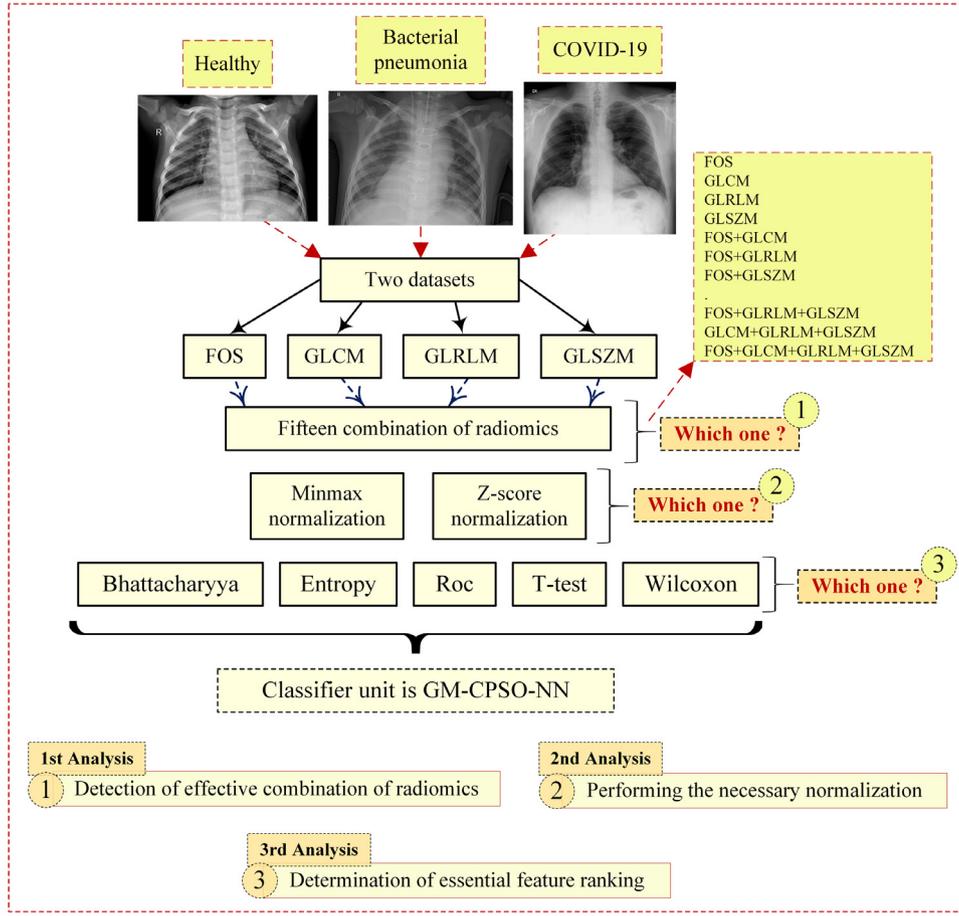


Fig. 2. General scheme of framework analysis.

entropy of each feature is determined from (51) by using mean values and standard deviations [27,37].

$$e = \frac{1}{2} \left[\left(\frac{\sigma_1^2}{\sigma_2^2} + \frac{\sigma_2^2}{\sigma_1^2} - 2 \right) + \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \right) (\mu_1 - \mu_2)^2 \right] \quad (51)$$

The *t*-test (*t*) score examines the mean values of two samples and focuses on the difference between them. The *t* score is obtained by using the mean, standard deviation, and sample sizes (n_1, n_2), as shown in (52) [26]. The absolute value of *t* is used to sort features. Larger values indicate higher importance, similar to the entropy test [37].

$$t = \mu_1 - \mu_2 \Big/ \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}} \quad (52)$$

The ROC curve is obtained using the tail functions, defined as $T_i(x)$, and it is equal to $(1 - F_i(x))$ for $i = 1, 2$. $F_i(x)$ denotes the distribution functions of X , and $F_1(x)$ and $F_2(x)$ denote the distribution functions of the two classes. The ROC is defined as follows [37].

$$ROC(t) = T_1(T_2^{-1}(t)), \quad t \in 0, 1 \quad (53)$$

The Wilcoxon test is also termed Mann Whitney *U* test, and it is used to calculate the feature importance via medians. It can assign the null hypothesis in the event that normal distribution is not observed. If *R* denotes the sum of ranks of samples, the *z* score being mentioned for ranking, is obtained as follows [27].

$$z = R - \frac{n_1(n_1 + n_2 + 1)}{2} \Big/ \sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}} \quad (54)$$

2.4. General scheme of framework analysis

In this paper, a comprehensive framework is proposed for COVID-19 discrimination. X-ray images were first obtained from different databases to form two data sets. FOS, GLCM, GLRLM, and GLSZM features were extracted from images to determine the radiomics to be used. Selective information was acquired by considering the necessary combination of the radiomics. Fifteen sub-data sets with different radiomic(s) were evaluated to ascertain the necessary combination. Two normalization approaches (*minmax* and *z-score*) were also tested to identify the appropriate one for the framework. After the identification of the necessary radiomics and normalization approach, five feature ranking methods (*Bhattacharyya*, *entropy*, *Roc*, *t-test*, and *Wilcoxon*) were used to determine the classification process for which the best performance of the framework was achieved. All evaluations were performed with GM-CPSO-NN that proved oneself for the classification of different medical data sets. Fig. 2 shows the general scheme of framework analysis.

As evident in Fig. 2, data normalization was performed after radiomics were obtained. Data combination analysis was performed together with the normalization technique. Subsequently, to the data that included the required radiomics combination and normalized with the appropriate technique, feature ranking was applied to finalize the COVID-19 framework of which GM-CPSO-NN is kept as the classifier unit.

3. Experimental results and interpretations

The first part of this section presents the features of the data used, and the second part discusses an experimental analysis and interpretations. Experiments were examined in three categories: (1) detection

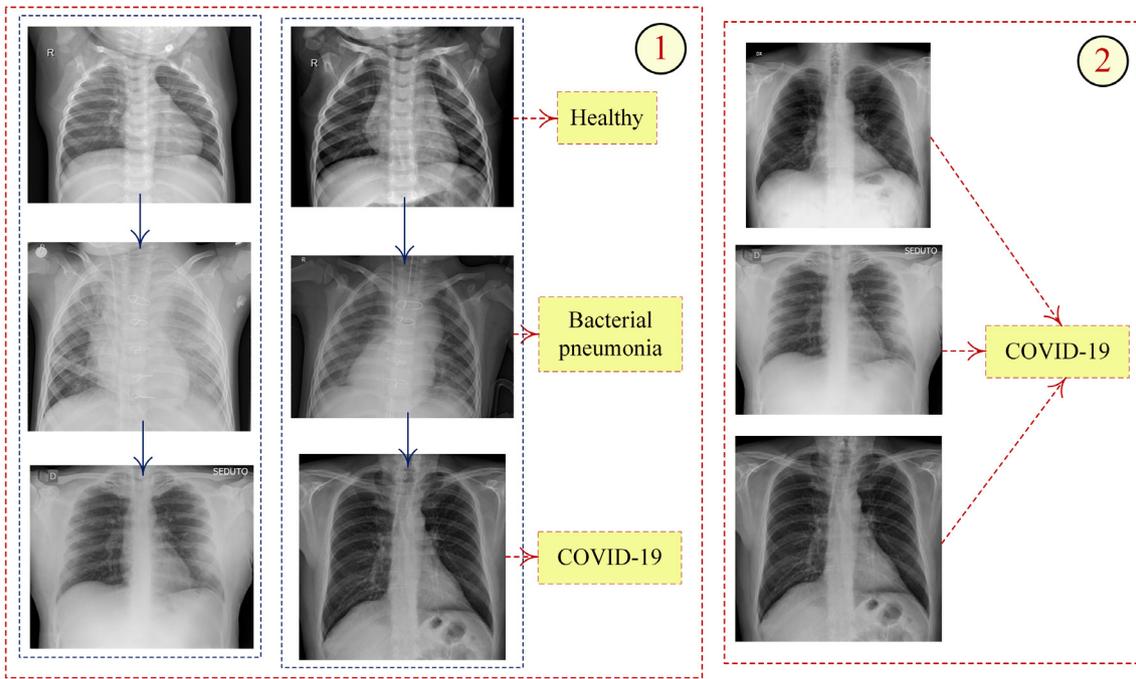


Fig. 3. Handicaps of data sets.

of the necessary combination of effective radiomics, (2) identification of the appropriate normalization approach, (3) determination of the essential feature ranking method. Seven metrics (accuracy, AUC, sensitivity, specificity, g-mean, precision, and f-measure) were used to comprehensively evaluate the framework performance [10,17]; their equations are presented in (55)–(61). Twofold cross validation was used as the test method since it offers a challenging approach to evaluate the frameworks and to assess the imbalanced data sets. For details on the determination of the metrics, the reader is referred to Ref. [28].

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (55)$$

$$AUC = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n 1_{p_i > p_j} \quad (56)$$

$$Sensitivity = \frac{TP}{(TP + FN)} \quad (57)$$

$$Specificity = \frac{TN}{(TN + FP)} \quad (58)$$

$$G - mean = \sqrt{Sensitivity \times Specificity} \quad (59)$$

$$Precision = \frac{TP}{(TP + FP)} \quad (60)$$

$$F - measure = 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity} \quad (61)$$

3.1. Features of data sets

The data used consisted of X-ray images with normal, bacterial pneumonia, and coronavirus labels. To construct a task-affirmative framework, we used a classification process in which normal and bacterial pneumonia were considered as negative classes and COVID-19 was treated as a positive class. Two different databases were used to obtain a detailed data repository [38–40].

In our repository, the first data set (data #1) was constructed using 80 samples for all classes. The second data set (data #2) comprised 400 images, with 80 COVID-19, 160 bacterial pneumonia, and 160 normal images. The second data set was constructed to enforce the classification framework, and the data part involving coronavirus samples remained low in contrast to the normal and pneumonia samples.

Table 3

Parameter settings and examination for GM-CPSO-NN.

Parameter	Value/Range
Position ranges of weight $\rightarrow [X_{min}, X_{max}]$	[-10,10]
Position ranges of bias $\rightarrow [BX_{min}, BX_{max}]$	[-10,10]
Velocity ranges of weight $\rightarrow [V_{min}, V_{max}]$	[-2,2]
Velocity ranges of bias $\rightarrow [BV_{min}, BV_{max}]$	[-2,2]
Inertia weight	Gauss map
Population size	Variable in [5,10,...,100]
$c_1 = 4 - c_2$	Variable in [1.9,1.92,...,2.1]
Maximum iteration number	Variable in [500,1000,...,5000]
Hidden node number	Variable in [2,4,..., feature number *3]

In other words, there existed a data imbalance between positive and negative cases, and it rendered the task of finding positive class labels challenging; that is why this data set is mentioned [38–40]. This process revealed the efficiency of our framework and showed whether it could perform the necessary task under both conditions. There were two handicaps in the X-ray images used (Fig. 3).

1. The first handicap was the similar contrast and gray level features in X-ray images involving different class labels.
2. The second disadvantage arose because of the different contrast and gray level features among X-ray images belonging to the same class labels.

The framework should accurately distinguish different labels even if they have similar features. Furthermore, it should determine the same class labels in the event that they have different characteristics. Though an efficient classifier (GM-CPSO-NN) was used to classify the challenging data set, feature analyses was the most important examination that could directly increase the classification success for a customized-COVID-19 framework.

Table 4
Performance comparison in terms of radiomics for data #1 (minmax normalization is active).

Data features/Success rates	Accuracy	AUC	Sensitivity	Specificity	G-mean	Precision	F-measure
FOS	98,33	98,75	100	97,50	98,74	95,24	97,56
GLCM	93,33	90,94	83,75	98,13	90,65	95,71	89,33
GLRLM	78,75	73,44	57,50	89,38	71,69	73,02	64,34
GLSZM	95,00	94,06	91,25	96,88	94,02	93,59	92,41
FOS + GLCM	97,92	98,13	98,75	97,50	98,12	95,18	96,93
FOS + GLRLM	98,33	98,75	100	97,50	98,74	95,24	97,56
FOS + GLSZM	97,92	98,13	98,75	97,50	98,12	95,18	96,93
GLCM + GLRLM	91,25	88,44	80,00	96,88	88,03	92,75	85,91
GLCM + GLSZM	94,58	92,50	86,25	98,75	92,29	97,18	91,39
GLRLM + GLSZM	96,25	95,63	93,75	97,50	95,61	94,94	94,34
FOS + GLCM + GLRLM	97,92	98,44	100	96,88	98,43	94,12	96,97
FOS + GLCM + GLSZM	96,67	96,88	97,50	96,25	96,87	92,86	95,12
FOS + GLRLM + GLSZM	97,92	97,81	97,50	98,13	97,81	96,30	96,89
GLCM + GLRLM + GLSZM	96,25	96,56	97,50	95,63	96,56	91,76	94,55
FOS + GLCM + GLRLM + GLSZM	97,08	95,94	92,50	99,38	95,88	98,67	95,48

(Optimum results are italicized and bolded for every feature combination.)

Table 5
Performance comparison in terms of radiomics for data #1 (z-score normalization is active).

Data features	Accuracy	AUC	Sensitivity	Specificity	G-mean	Precision	F-measure
FOS	97,92	98,44	100	96,88	98,43	94,12	96,97
GLCM	89,58	87,50	81,25	93,75	87,28	86,67	83,87
GLRLM	78,75	72,19	52,50	91,88	69,45	76,36	62,22
GLSZM	94,58	93,13	88,75	97,50	93,02	94,67	91,61
FOS + GLCM	95,42	94,38	91,25	97,50	94,32	94,81	92,99
FOS + GLRLM	97,50	98,13	100	96,25	98,11	93,02	96,39
FOS + GLSZM	96,67	97,19	98,75	95,63	97,17	91,86	95,18
GLCM + GLRLM	91,25	88,13	78,75	97,50	87,62	94,03	85,71
GLCM + GLSZM	94,58	94,69	95,00	94,38	94,69	89,41	92,12
GLRLM + GLSZM	96,67	96,25	95,00	97,50	96,24	95,00	95,00
FOS + GLCM + GLRLM	95,83	95,63	95,00	96,25	95,62	92,68	93,83
FOS + GLCM + GLSZM	96,25	94,69	90,00	99,38	94,57	98,63	94,12
FOS + GLRLM + GLSZM	98,75	99,06	100	98,13	99,06	96,39	98,16
GLCM + GLRLM + GLSZM	95,42	94,38	91,25	97,50	94,32	94,81	92,99
FOS + GLCM + GLRLM + GLSZM	94,58	94,69	95,00	94,38	94,69	89,41	92,12

(Optimum results are italicized and bolded for every feature combination.)

3.2. Performance evaluation and analysis

To increase the classification success and perform a detailed performance comparison, we comprehensively determined the parameter settings of GM-CPSO-NN, and they are presented in Table 3. The velocity ranges of the weight-bias values were maintained at 0.2 times the position ranges for achieving higher performance [27–29]. Four parameters (acceleration constants, hidden node number, population size, and maximum iteration number) were varied to observe the highest scores in trials by using seven metric-based comparisons. The necessary feature combination, reliable normalization, and appropriate feature ranking were determined by calculating the number of best scores on the seven metrics.

3.2.1. Results and analysis for data #1

Tables 4 and 5 show the performance comparison of feature combinations (radiomics evaluation) to which *minmax* and *z-score* approaches were applied. In the absence of the application of a feature ranking method, the first comparison (Tables 4 and 5) is related to the identification of effective radiomics. Subsequently, normalization analyses were performed.

Table 4 shows that the best success rates of five metrics (accuracy, AUC, sensitivity, f-measure, and g-mean) were achieved by using FOS and FOS + GLRLM features, and the scores are identical for the two features. If the accuracy and AUC metrics are scrutinized, five features (FOS, FOS + GLCM, FOS + GLRLM, FOS + GLCM + GLRLM, and FOS + GLRLM + GLSZM) are observed to yield success scores over 97% for data #1. The key point of the analyses is FOS features, and the addition of other features (except GLRLM) cannot provide higher performance. Even if the combination FOS + GLRLM reveals the same

results and cannot enhance the performance, it can also be considered as an appropriate preference.

Table 5 shows that the highest scores of five metrics (accuracy, AUC, sensitivity, f-measure, and g-mean) were observed for FOS + GLRLM + GLSZM features. Scores above 97% for the accuracy and AUC metrics were attained in only three conditions (FOS, FOS + GLRLM, and FOS + GLRLM + GLSZM). Although the choice of FOS + GLRLM provides a score above 97% for the accuracy and AUC metrics, it decreases the performance on six metrics according to the rates with FOS features. FOS + GLRLM + GLSZM features are superior in terms of attaining remarkable scores. Moreover, it can be inferred that some additions (GLRLM with FOS features) can decrease the performance while some combinations (FOS + GLRLM + GLSZM) can offer higher performance despite adding new features (GLSZM) to a performance-reducing preference (FOS + GLRLM).

Fig. 4 presents a summary of the best results assessed by using the *minmax* and *z-score* approaches for the FOS, FOS + GLRLM, and FOS + GLRLM + GLSZM features. The best results were obtained on six metrics by applying *z-score* to FOS + GLRLM + GLSZM features. Although *z-score*-based results are higher than *minmax*-based scores, the rates are very close.

In the third part, feature ranking methods were used after *minmax* was applied to FOS features and *z-score* was implemented for FOS + GLRLM + GLSZM features for achieving the maximum performance. Herein, proposed framework will take shape according to this comparison. Tables 6 and 7 show a comparison of feature ranking methods used after *minmax* and *z-score* were applied to FOS and FOS + GLRLM + GLSZM features, respectively.

In Table 6, the highest accuracy (99.17%), AUC (99.06%), g-mean (99.06%), and f-measure (98.75%) scores are observed for *Bhat-tacharyya* and *Roc* rankings. In Table 7, results showing the highest

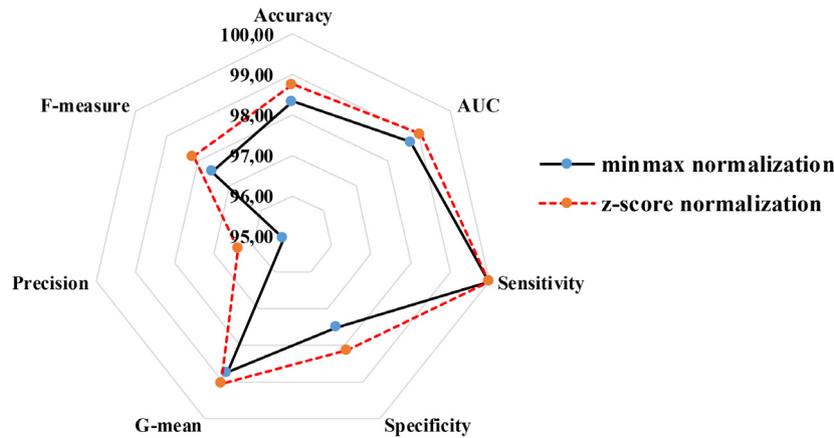


Fig. 4. Comparison of normalization approaches for data #1.

Table 6 Comparison of feature ranking methods for data #1 (FOS and minmax).

Feature ranking method	Accuracy	AUC	Sensitivity	Specificity	G-mean	Precision	F-measure
Bhattacharyya	99,17	99,06	98,75	99,38	99,06	98,75	98,75
Entropy	98,75	99,06	100	98,13	99,06	96,39	98,16
Roc	99,17	99,06	98,75	99,38	99,06	98,75	98,75
T-test	99,17	98,75	97,5	100	98,74	100	98,73
Wilcoxon	98,33	98,75	100	97,50	98,74	95,24	97,56

(Optimum results are italicized and bolded for every feature combination.)

Table 7 Comparison of feature ranking methods for data #1 (FOS + GLRLM + GLSZM and z-score).

Feature ranking method	Accuracy	AUC	Sensitivity	Specificity	G-mean	Precision	F-measure
Bhattacharyya	98,75	98,75	98,75	98,75	98,75	97,53	98,14
Entropy	98,33	98,75	100	97,50	98,74	95,24	97,56
Roc	97,92	98,44	100	96,88	98,43	94,12	96,97
T-test	98,75	99,06	100	98,13	99,06	96,39	98,16
Wilcoxon	98,33	98,75	100	97,50	98,74	95,24	97,56

(Optimum results are italicized and bolded for every feature combination.)

accuracy, AUC, sensitivity, g-mean, and f-measure rates are observed for *t-test* ranking. Fig. 5 is presented to provide a detailed view and an objective assessment of the best results of two frameworks.

As evident in Fig. 5, both preferences achieve results close to each other. However, if the first framework (FOS and *minmax* and *Bhattacharyya* or *Roc*) is chosen, the highest accuracy (99.17%) can be achieved besides better scores on specificity, precision, and f-measure metrics. Thus, the aforementioned feature combination, normalization approach, and ranking method were the best choice for data #1-based experiments.

3.2.2. Results and analysis for data #2

Tables 8 and 9 present performance evaluations of feature combinations to which *minmax* and *z-score* approaches have been applied. In the absence of feature ranking, the first comparison (Tables 8 and 9) shows the effectiveness of radiomic combinations. The necessary normalization method is examined next.

In Table 8, FOS + GLRLM features show the highest accuracy (99%), specificity (99.38%), precision (97.50%) and f-measure (97.50%) scores, while FOS features yield the best results for the AUC, sensitivity, and g-mean metrics. With regard to the importance and rank of the first places on the basis of success rates, FOS + GLRLM features show high performance for the classification of data #2. It is apparent that FOS features are necessary for achieving the best performances, either in combination with other features or by itself. Five conditions (FOS, FOS + GLCM, FOS + GLRLM, FOS + GLCM + GLSZM, and FOS + GLRLM + GLSZM) showed a success rate above 97% in terms of the accuracy and AUC rates. However, the use of FOS + GLRLM features showed higher accuracy.

In Table 9, the best accuracy (99%), specificity (99.38%), precision (97.50%), and f-measure (97.50%) rates are attained by FOS + GLRLM features. Furthermore, FOS + GLSZM features yield the highest scores for the AUC, sensitivity, and g-mean metrics. For four features (FOS, FOS + GLCM, FOS + GLRLM, and FOS + GLSZM), a success rate over 97% was achieved for the accuracy and AUC metrics. However, FOS + GLRLM features can help increase the accuracy rate. Consequently, FOS + GLRLM features remain the best choice in both experiments in which *minmax* or *z-score* is applied.

If the highest scores of FOS + GLRLM and *minmax* and FOS + GLRLM and *z-score* preferences are examined, it is observed that all scores are identical for data #2. Hence, both normalization methods can be preferred for application to FOS + GLRLM features.

In the third part, feature ranking methods were tested on the combination of FOS + GLRLM and *minmax* and on FOS + GLRLM and *z-score*. Tables 10 and 11 show the comparisons for data #2. Herein, general framework design takes shape according to this comparison besides the consequences in Section 3.2.1.

In Table 10, the highest scores on all metrics are observed if the feature ranking approach is chosen as *Bhattacharyya*, *entropy*, or *Roc* for FOS + GLRLM and *minmax* preference. In Table 11, the best scores are obtained only with *t-test* feature ranking for FOS + GLRLM and *z-score* preference. Fig. 6 presents a comprehensive view of the best results of the two frameworks.

As evident in Fig. 6, even if both preferences achieve very similar results, the first choice (FOS + GLRLM and *minmax* and *Bhattacharyya*, *entropy*, or *Roc*) is superior to the second one, and it provides the highest performances on six metrics. Consequently, the aforementioned feature combination, normalization approach, and ranking method were

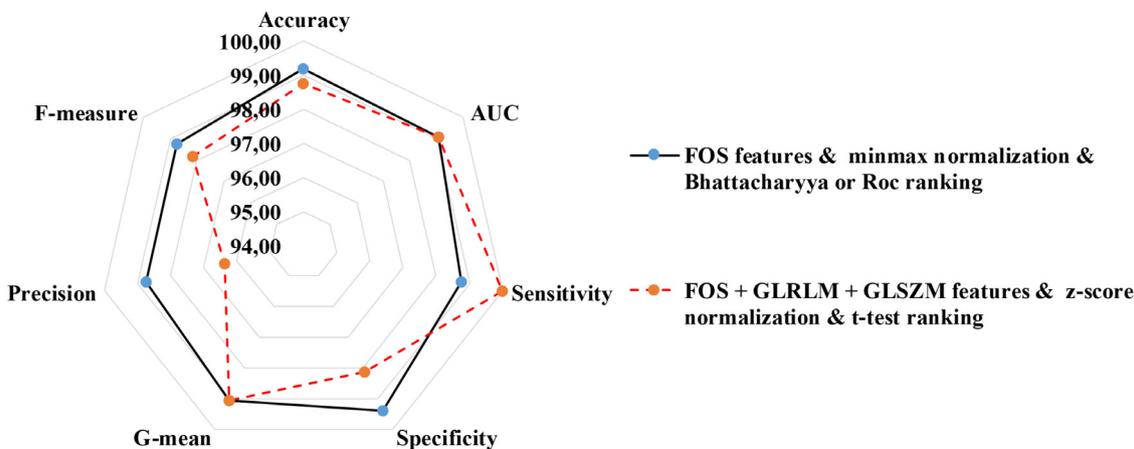


Fig. 5. Comparison of two frameworks for data #1.

Table 8

Performance comparison in terms of radiomics for data #2 (minmax normalization is active).

Data features/Success rates	Accuracy	AUC	Sensitivity	Specificity	G-mean	Precision	F-measure
FOS	98,75	99,22	100	98,44	99,22	94,12	96,97
GLCM	93,75	87,66	77,50	97,81	87,07	89,86	83,22
GLRLM	86,75	70,63	43,75	97,50	65,31	81,40	56,91
GLSZM	95,00	91,25	85,00	97,50	91,04	89,47	87,18
FOS + GLCM	98,25	97,03	95,00	99,06	97,01	96,20	95,60
FOS + GLRLM	99,00	98,44	97,50	99,38	98,43	97,50	97,50
FOS + GLSZM	98,00	96,88	95,00	98,75	96,86	95,00	95,00
GLCM + GLRLM	94,25	90,31	83,75	96,88	90,07	87,01	85,35
GLCM + GLSZM	95,25	92,81	88,75	96,88	92,72	87,65	88,20
GLRLM + GLSZM	94,75	89,69	81,25	98,13	89,29	91,55	86,09
FOS + GLCM + GLRLM	97,75	95,31	91,25	99,38	95,23	97,33	94,19
FOS + GLCM + GLSZM	97,75	97,66	97,50	97,81	97,66	91,76	94,55
FOS + GLRLM + GLSZM	98,50	97,19	95,00	99,38	97,16	97,44	96,20
GLCM + GLRLM + GLSZM	94,50	89,53	81,25	97,81	89,15	90,28	85,53
FOS + GLCM + GLRLM + GLSZM	97,50	95,16	91,25	99,06	95,08	96,05	93,59

(Optimum results are italicized and bolded for every feature combination.)

Table 9

Performance comparison in terms of radiomics for data #2 (z-score normalization is active).

Data Features	Accuracy	AUC	Sensitivity	Specificity	G-mean	Precision	F-measure
FOS	98,50	98,59	98,75	98,44	98,59	94,05	96,34
GLCM	93,75	88,59	80,00	97,19	88,18	87,67	83,66
GLRLM	86,75	72,97	50,00	95,94	69,26	75,47	60,15
GLSZM	95,25	92,34	87,50	97,19	92,22	88,61	88,05
FOS + GLCM	98,50	98,59	98,75	98,44	98,59	94,05	96,34
FOS + GLRLM	99,00	98,44	97,50	99,38	98,43	97,50	97,50
FOS + GLSZM	98,75	98,75	98,75	98,75	98,75	95,18	96,93
GLCM + GLRLM	94,75	89,22	80,00	98,44	88,74	92,75	85,91
GLCM + GLSZM	94,50	90,00	82,50	97,50	89,69	89,19	85,71
GLRLM + GLSZM	94,75	91,56	86,25	96,88	91,41	87,34	86,79
FOS + GLCM + GLRLM	96,75	95,16	92,50	97,81	95,12	91,36	91,93
FOS + GLCM + GLSZM	96,50	93,13	87,50	98,75	92,95	94,59	90,91
FOS + GLRLM + GLSZM	97,25	95,00	91,25	98,75	94,93	94,81	92,99
GLCM + GLRLM + GLSZM	96,00	93,28	88,75	97,81	93,17	91,03	89,87
FOS + GLCM + GLRLM + GLSZM	97,50	95,63	92,50	98,75	95,57	94,87	93,67

(Optimum results are italicized and bolded for every feature combination.)

Table 10

Comparison of feature ranking methods for data #2 (FOS + GLRLM and minmax).

Feature ranking method	Accuracy	AUC	Sensitivity	Specificity	G-mean	Precision	F-measure
Bhattacharyya	99,25	99,53	100	99,06	99,53	96,39	98,16
Entropy	99,25	99,53	100	99,06	99,53	96,39	98,16
Roc	99,25	99,53	100	99,06	99,53	96,39	98,16
T-test	99,00	99,38	100	98,75	99,37	95,24	97,56
Wilcoxon	99,00	99,38	100	98,75	99,37	95,24	97,56

(Optimum results are italicized and bolded for every feature combination.)

Table 11
Comparison of feature ranking methods for data #2 (FOS + GLRLM and z-score).

Feature ranking method	Accuracy	AUC	Sensitivity	Specificity	G-mean	Precision	F-measure
Bhattacharyya	98,75	99,22	100	98,44	99,22	94,12	96,97
Entropy	98,75	99,22	100	98,44	99,22	94,12	96,97
Roc	98,75	99,22	100	98,44	99,22	94,12	96,97
T-test	99,00	99,38	100	98,75	99,37	95,24	97,56
Wilcoxon	98,75	99,22	100	98,44	99,22	94,12	96,97

(Optimum results are italicized and bolded for every feature combination.)

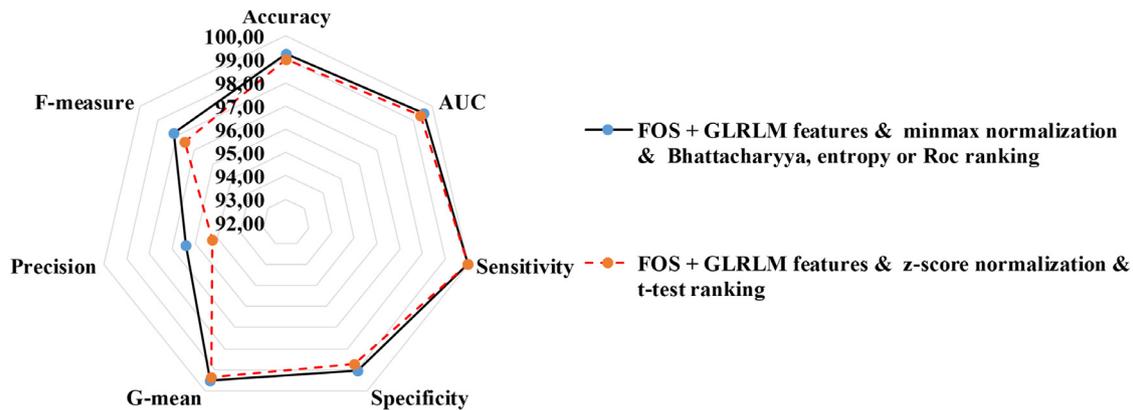


Fig. 6. Comparison of two frameworks for data #2.

the most appropriate adjustment, ensuring remarkable performance in data #2 based experiments (e.g., 99.25% accuracy and 99.53% AUC).

4. Discussions

From the analyses of Section 3, the following inferences can be drawn.

- As in the study of Koyuncu et al. [27] pertaining to brain tumor classification in MR images, it was found that FOS features are necessary for the construction of a task-affirmative framework and to discriminate coronavirus in X-ray images. In other words, FOS features are necessary to construct a specified-coronavirus framework for X-ray imaging modality.
- Among the radiomic combinations, FOS and FOS + GLRLM features can be chosen for data including normal and imbalanced data distributions, respectively.
- The *minmax* approach can be preferred as the normalization approach if it is used with radiomics and feature ranking methods.
- For the highest performance, *Bhattacharyya* and *Roc* rankings seem remarkable on both data sets formed with necessary radiomics to which *minmax* normalization has been applied.
- In this study, GM-CPSO-NN proves its efficiency again as in [27–29,31] and can be chosen as a state-of-the-art classifier.

Our framework was designed according to the deductions, and it is presented in Fig. 7. It involves FOS or FOS + GLRLM features as the radiomic features, *minmax* as the normalization approach, *Bhattacharyya* or *Roc* rankings as the feature selection method, and GM-CPSO-NN as the classifier unit.

As declared before, data #1 has a normal distribution while data #2 is specified as an imbalanced data set. At this juncture, the classification performance for data #1 is expected to be higher than the results for data #2 since the detection of coronavirus patterns of data #2 is more challenging owing to the smaller number of samples for the COVID-19 label. However, our framework achieved significant scores for data #1,

especially for the accuracy (99.17%) and AUC (99.06%) metrics. It also showed better performance for data #2 by presenting success scores of 99.25% and 99.53% for the accuracy and AUC metrics, respectively. These results show the robustness of the proposed framework, implying that it can be considered for use with both normal and imbalanced data sets.

As evident in the framework, three parts, namely, normalization, feature ranking, and classifier, remain the same for both data set conditions. Only the feature extraction (radiomics) method differs with the data set handled. In other words, necessary radiomics are considered according to the data used. For real-time applications, if a designer has balanced data and proposes to use our supervised-framework, it is suggested that FOS features be used as radiomics. If the designer has an imbalanced data set and intends to use the framework, it is recommended that FOS + GLRLM features be used as the necessary radiomics so as to achieve the highest performance. After the training of the classifier in the framework, the test or real-time applications can be implemented by selecting the best network found during training.

Apart from the seven metric-based comparisons, the computation time analysis of the framework was evaluated. The training–test time and test time were observed separately for both data sets. For data #1, the proposed framework resulted in a training–test time and a test time of 1.7316 s and 0.0031 s, respectively. For data #2, these times were 2.5425 s and 0.0039 s, respectively. Clearly, our framework achieves remarkable results in a very short time, especially for the test that directly provides the operation time.

Table 12 presents a comparison of our framework with that proposed by state-of-the-art studies. The proposed framework shows remarkable performance when compared with transfer-learning-based models, deep-learning-based pipelines, specific deep learning methods, and more complex systems comprising them. Accuracy and AUC rates achieved by the proposed framework are significant and yield appreciable performance vs. other studies. This superiority can be better observed in an accuracy-based comparison, especially for the results of studies involving binary classification.

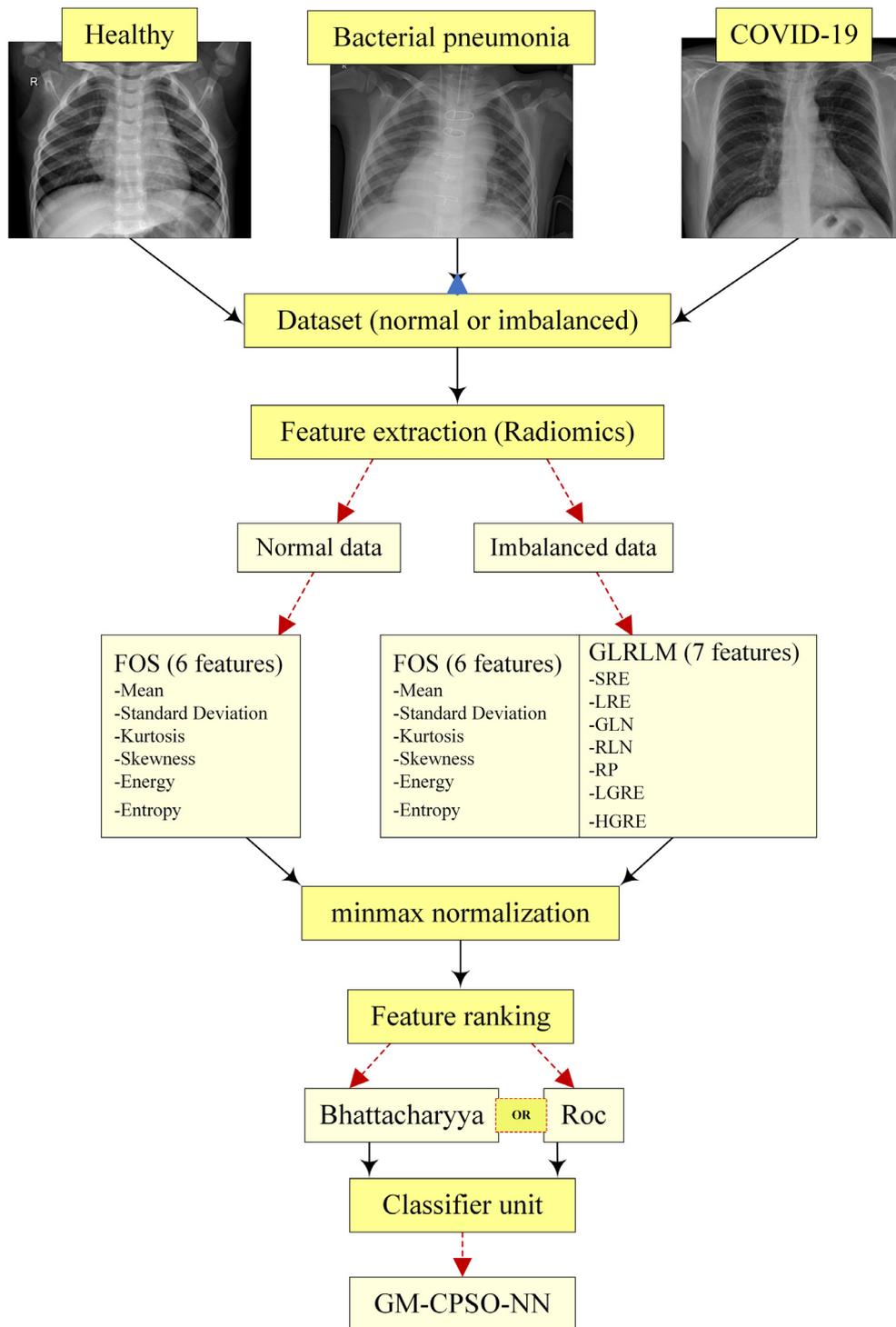


Fig. 7. Flowchart of the proposed framework.

5. Conclusions

In this paper, a robust framework is presented to identify COVID-19 in X-ray images. The radiomics priority, normalization preferences, and active feature rankings were examined together to design the specified-coronavirus framework equipped with a state-of-the-art classifier.

In experiments, remarkable results were obtained, especially the accuracy and AUC metrics for both (normal and imbalanced) data sets. It is concluded that the preferences including FOS and FOS + GLRLM features normalized with the *minmax* approach to which *Bhattacharyya*

or *Roc* rankings are applied can be considered to distinguish COVID-19 patterns in normal- and imbalanced-data conditions. FOS features showed high efficiency and were found to be decisive for COVID-19 detection in X-ray images. Furthermore, in the design of a robust model, apart from the determination of appropriate feature extraction (radiomics and normalization evaluation), feature ranking is also an important factor for achieving the highest performance.

In a future work, we intend to test the applicability of our framework to CT-based coronavirus detection. We intend to test its efficiency for both normal and imbalanced data sets.

Table 12
Comparison of results of the present study with those of previous studies.

Study	Imaging Modality	Data set	Method	Task	Results (%)	
					Accuracy	AUC
Ozturk et al. [6]	X-ray	125 COVID-19/500 no finding	DarkCovidNet model with CNN structure	COVID-19 vs. no finding	98.08	–
Ozturk et al. [6]	X-ray	125 COVID-19/500 no finding/500 pneumonia	DarkCovidNet model with CNN structure	COVID-19 vs. no finding vs.pneumonia	87.02	–
Toğaçar et al. [7]	X-ray	295 COVID-19/65 normal/98 pneumonia	Model performing feature extraction with deep learning, optimization with SMO, classification with SVM	COVID-19 vs. normal vs.pneumonia	99.27	–
Ucar and Korkmaz [8]	X-ray	79 COVID-19/1583 normal/4290 pneumonia	Offline data augmentation, Bayesian optimization during DeepSqueezeNet training	COVID-19 vs. normal vs. pneumonia	98.26	–
Apostolopoulos and Mpesiana [9]	X-ray	224 COVID-19/504 normal/700 bacterial pneumonia	Transfer learning with VGG19	COVID-19 vs. others	98.75	–
Apostolopoulos and Mpesiana [9]	X-ray	224 COVID-19/504 normal/700 bacterial pneumonia	Transfer learning with VGG19	COVID-19 vs. normal vs. bacterial pneumonia	93.48	–
Apostolopoulos and Mpesiana [9]	X-ray	224 COVID-19/504 normal/714 pneumonia (bacterial and viral)	Transfer learning with MobileNet-v2	COVID-19 vs. others	96.78	–
Apostolopoulos and Mpesiana [9]	X-ray	224 COVID-19/504 normal/714 pneumonia (bacterial and viral)	Transfer learning with MobileNet-v2	COVID-19 vs. normal vs. pneumonia	94.72	–
Butt et al. [10]	CT	184 COVID-19/145 normal/194 influenza-A viral pneumonia	Model segmented the candidate region with 3D-CNN and classified using Noisy-OR Bayesian function	COVID-19 vs. others	–	99.6
Li et al. [11]	CT	1296 COVID-19/1735 CAP/1325 nonpneumonia	Model used ResNet50 for classification	COVID-19 vs. others	–	96
Kang et al. [12]	CT	1495 COVID-19/1027 pneumonia	Model used VNet to segment the lesion region, extracted radiomic and hand-crafted features, and used five different classifiers for classification	COVID-19 vs. pneumonia	95.5	–
Afshar et al. [13]	X-ray	123 COVID-19/1341 normal/3845 pneumonia (bacterial and viral)	Model with capsule network structure	COVID-19 vs. normal vs. pneumonia	98.3	97
Mahdy et al. [14]	X-ray	25 COVID-19/15 normal	Multilevel thresholding and classification with SVM	COVID-19 vs. normal	97.48	–
Hemdan et al. [15]	X-ray	25 COVID-19/25 normal	Model with different deep learning structures	COVID-19 vs. normal	90	90
This study	X-ray	80 COVID-19/80 normal/80 bacterial pneumonia	Framework involved radiomics, normalization, feature ranking, and GM-CPSO–NN	COVID-19 vs. others	99.17	99.06
This study	X-ray	80 COVID-19/160 normal/160 bacterial pneumonia	Framework involved radiomics, normalization, feature ranking, and GM-CPSO–NN	COVID-19 vs. others	99.25	99.53

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] C.-C. Lai, T.-P. Shih, W.-C. Ko, H.-J. Tang, P.-R. Hsueh, Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) and corona virus disease-2019 (COVID-19): the epidemic and the challenges, *Int. J. Antimicrob. Ag.* (2020) 105924.
- [2] S. Kang, W. Peng, Y. Zhu, S. Lu, M. Zhou, W. Lin, W. Wu, S. Huang, L. Jiang, X. Luo, Recent progress in understanding 2019 novel coronavirus associated with human respiratory disease: detection, mechanism and treatment, *Int. J. Antimicrob. Ag.* (2020) 105950.
- [3] D. Singh, V. Kumar, M. Kaur, Classification of COVID-19 patients from chest CT images using multi-objective differential evolution-based convolutional neural networks, *Eur. J. Clin. Microbiol. Infect. Dis.* (2020) 1–11.
- [4] K. Li, Y. Fang, W. Li, C. Pan, P. Qin, Y. Zhong, X. Liu, M. Huang, Y. Liao, S. Li, CT image visual quantitative evaluation and clinical classification of coronavirus disease (COVID-19), *Eur. Radiol.* (2020) 1–10.
- [5] S.J. Fong, G. Li, N. Dey, R.G. Crespo, E. Herrera-Viedma, Composite Monte Carlo decision making under high uncertainty of novel coronavirus epidemic using hybridized deep learning and fuzzy rule induction, *Appl. Soft Comput.* (2020) 106282.
- [6] T. Ozturk, M. Talo, E.A. Yildirim, U.B. Baloglu, O. Yildirim, U.R. Acharya, Automated detection of COVID-19 cases using deep neural networks with X-ray images, *Comput. Biol. Med.* (2020) 103792.
- [7] M. Toğaçar, B. Ergen, Z. Cömert, COVID-19 detection using deep learning models to exploit social mimic optimization and structured chest X-ray images using fuzzy color and stacking approaches, *Comput. Biol. Med.* (2020) 103805.
- [8] F. Ucar, D. Korkmaz, Covidiagnosis-net: Deep Bayes-squeezenet based diagnostic of the coronavirus disease 2019 (COVID-19) from X-ray images, *Med. Hypotheses* (2020) 109761.
- [9] I.D. Apostolopoulos, T.A. Mpesiana, Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks, *Phys. Eng. Sci. Med.* (2020) 1.
- [10] C. Butt, J. Gill, D. Chun, B.A. Babu, Deep learning system to screen coronavirus disease 2019 pneumonia, *Appl. Intell.* (2020) 1.
- [11] L. Li, L. Qin, Z. Xu, Y. Yin, X. Wang, B. Kong, J. Bai, Y. Lu, Z. Fang, Q. Song, Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT, *Radiology* (2020) 200905.
- [12] H. Kang, L. Xia, F. Yan, Z. Wan, F. Shi, H. Yuan, H. Jiang, D. Wu, H. Sui, C. Zhang, Diagnosis of coronavirus disease 2019 (covid-19) with structured latent multi-view representation learning, *IEEE Trans. Med. Imaging* (2020).
- [13] P. Afshar, S. Heidarian, F. Naderkhani, A. Oikonomou, K.N. Plataniotis, A. Mohammadi, Covid-caps: A capsule network-based framework for identification of covid-19 cases from x-ray images, 2020, arXiv preprint arXiv:2004.02696.

- [14] L.N. Mahdy, K.A. Ezzat, H.H. Elmousalami, H.A. Ella, A.E. Hassanien, Automatic x-ray covid-19 lung image classification system based on multi-level thresholding and support vector machine, 2020, medRxiv.
- [15] E.E.-D. Hemdan, M.A. Shouman, M.E. Karar, Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in x-ray images, 2020, arXiv preprint arXiv:2003.11055.
- [16] L. Yan, H.-T. Zhang, J. Goncalves, Y. Xiao, M. Wang, Y. Guo, C. Sun, X. Tang, L. Jin, M. Zhang, A machine learning-based model for survival prediction in patients with severe COVID-19 infection, 2020, MedRxiv.
- [17] J. Wu, P. Zhang, L. Zhang, W. Meng, J. Li, C. Tong, Y. Li, J. Cai, Z. Yang, J. Zhu, Rapid and accurate identification of COVID-19 infection through machine learning based on clinical available blood test results, 2020, medRxiv.
- [18] C. Feng, Z. Huang, L. Wang, X. Chen, Y. Zhai, F. Zhu, H. Chen, Y. Wang, X. Su, S. Huang, A novel triage tool of artificial intelligence assisted diagnosis aid system for suspected COVID-19 pneumonia in fever clinics, 2020.
- [19] F. Yang, M. Hamit, C.B. Yan, J. Yao, A. Kutluk, X.M. Kong, S.X. Zhang, Feature extraction and classification on esophageal X-ray images of xinjiang kazak nationality, *J. Healthcare Eng.* 2017 (2017).
- [20] X. Liao, B. Cai, B. Tian, Y. Luo, W. Song, Y. Li, Machine-learning based radiogenomics analysis of MRI features and metagenes in glioblastoma multiforme patients with different survival time, *J. Cellular Mol. Med.* 23 (2019) 4375–4385.
- [21] H. Koyuncu, R. Ceylan, S. Asoglu, H. Cebeci, M. Koplay, An extensive study for binary characterisation of adrenal tumours, *Med. Biol. Eng. Comput.* 57 (2019) 849–862.
- [22] M. Barstugan, R. Ceylan, S. Asoglu, H. Cebeci, M. Koplay, Adrenal tumor characterization on magnetic resonance images, *Int. J. Imaging Syst. Technol.* 30 (2020) 252–265.
- [23] T. Yang, H. Zhu, X. Gao, Y. Zhang, Y. Hui, F. Wang, Grading of metacarpophalangeal rheumatoid arthritis on ultrasound images using machine learning algorithms, *IEEE Access* 8 (2020) 67137–67146.
- [24] U. Raghavendra, H. Fujita, A. Gudigar, R. Shetty, K. Nayak, U. Pai, J. Samanth, U.R. Acharya, Automated technique for coronary artery disease characterization and classification using DD-DTDWT in ultrasound images, *Biomed. Signal Process. Control* 40 (2018) 324–334.
- [25] M.Z. do Nascimento, A.S. Martins, T.A.A. Tosta, L.A. Neves, Lymphoma images analysis using morphological and non-morphological descriptors for classification, *Comput. Methods Programs Biomed.* 163 (2018) 65–77.
- [26] M. Momenzadeh, M. Sehhati, H. Rabbani, A novel feature selection method for microarray data classification based on hidden Markov model, *J. Biomed. Inform.* 95 (2019) 103213.
- [27] H. Koyuncu, M. Barstugan, U. Ozic, A comprehensive study about brain tumor discrimination by using feature ranking, mutual information and hybridized classifiers, *Med. Biol. Eng. Comput.* 58 (2020) 2971–2987.
- [28] H. Koyuncu, Gm-cps0: a new viewpoint to chaotic particle swarm optimization via gauss map, *Neural Process. Lett.* (2020).
- [29] H. Koyuncu, Parkinson's disease recognition using Gauss map based chaotic particle swarm-neural network, in: 2019 International Conference on Engineering and Telecommunication (EnT), IEEE, 2019, pp. 1–4.
- [30] S. Saremi, S. Mirjalili, A. Lewis, Biogeography-based optimisation with chaos, *Neural Comput. Appl.* 25 (2014) 1077–1097.
- [31] H. Koyuncu, Loss function selection in NN based classifiers: Try-outs with a novel method, in: 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), IEEE, 2020, pp. 1–4.
- [32] R.M. Haralick, K. Shanmugam, I.H. Dinstein, Textural features for image classification, *IEEE Trans. Syst. Man Cybern.* (1973) 610–621.
- [33] K.L.-C. Hsieh, C.-M. Lo, C.-J. Hsiao, Computer-aided grading of gliomas based on local and global MRI features, *Comput. Methods Programs Biomed.* 139 (2017) 31–38.
- [34] W. Gómez, W.C.A. Pereira, A.F.C. Infantosi, Analysis of co-occurrence texture statistics as a function of gray-level quantization for classifying breast ultrasound, *IEEE Trans. Med. Imaging* 31 (2012) 1889–1899.
- [35] A.S.M. Sohail, P. Bhattacharya, S.P. Mudur, S. Krishnamurthy, Local relative GLRLM-based texture feature extraction for classifying ultrasound medical images, in: 2011 24th Canadian Conference on Electrical and Computer Engineering (CCECE), IEEE, 2011, 001092-001095.
- [36] G. Thibault, J. Angulo, F. Meyer, Advanced statistical matrices for texture characterization: application to cell classification, *IEEE Trans. Biomed. Eng.* 61 (2013) 630–637.
- [37] T. Nguyen, A. Khosravi, D. Creighton, S. Nahavandi, A novel aggregate gene selection method for microarray data classification, *Pattern Recognit. Lett.* 60 (2015) 16–23.
- [38] D.S. Kermany, M. Goldbaum, W. Cai, C.C. Valentim, H. Liang, S.L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, Identifying medical diagnoses and treatable diseases by image-based deep learning, *Cell* 172 (2018) 1122–1131, e1129.
- [39] Kaggle, in, 2020.
- [40] SIRM, in, 2020.