International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 41, No. 4, August, 2024, pp. 1707-1726
Journal homepage: http://iieta.org/journals/ts

An Automated Computed Tomography Scan Analysis Framework for COVID-19 Detection Using Machine Learning

Sunil Dalal[1], Jyoti Prakash Singh[1*], Arvind Kumar Tiwari[2], Durgesh Nandan[3]

[1] National Institute of Technology Patna, Bihar 80006, India
[2] Kamla Nehru Institute of Technology, Sultanpur 228118, U.P., India
[3] School of CS & AI, S R University, Warangal 506371, Telangana, India

Corresponding Author Email: jps@nitp.ac.in

https://doi.org/10.18280/ts.410406

Received: 1 May 2023
Revised: 22 September 2023
Accepted: 30 October 2023
Available online: 31 August 2024

Keywords:
COVID-19 detection, computed tomography, Kernel extreme machine learning, feature extraction, classification, autoencoder, seagull optimization

ABSTRACT

During the coronavirus disease-19 (COVID-19) epidemic, there has been a growing need for rapid diagnostic tools, with Computed Tomography (CT) scans emerging as essential diagnostic resources. Nevertheless, the process of manually interpreting their findings, although informative, is nevertheless characterized by a significant amount of work and variability. In the current study, we intend to construct a machine learning-based model to automate the evaluation of CT images for COVID-19 diagnosis and to differentiate it from pneumonia and other non-COVID diseases. The model we propose employs a Tolerant Local Median Fuzzy C-means (TLMFCM) segmentation strategy in conjunction with the Stacked Sparse Autoencoder (SSAE) for robust feature extraction. The classification task employs a Locally Controlled Seagull Kernel Extreme Machine Learning (LCS-KELM) whose parameters are optimized with the Seagull Optimization algorithm (SOA). Our model performed better than other models in preliminary comparisons against traditional benchmarks, with an accuracy of 96.3% and a faster processing time.

1. INTRODUCTION

The emergence of a novel coronavirus in December 2019, subsequently named COVID-19, triggered a global health crisis, leading the World Health Organization (WHO) to declare it a public health emergency. COVID-19 primarily affects the respiratory system, encompassing a spectrum from mild to severe respiratory distress [1]. As of January 2021, this disease has afflicted over 90 million individuals across more than 200 countries, resulting in over 2 million fatalities as per the WHO. Early diagnosis is pivotal in mitigating the spread of COVID-19 and ensuring timely treatment. While reverse-transcription polymerase chain reaction (RT-PCR) is considered the gold standard for diagnosis, it possesses limitations, including time consumption, sample dependence, and operator variability [2]. As an alternative, chest X-rays (CXRs) and CT scans have been employed for initial detection, offering increased precision in identifying COVID-19-related abnormalities, especially in CT images [3, 4]. However, the ever-increasing volume of CT scan images poses a significant challenge to radiologists and physicians, leading to substantial delays in diagnosis [5].

The integration of biomedical image analysis with deep learning techniques (DLT) has emerged as a promising solution to expedite the interpretation of complex medical images [6, 7]. Radiographs reveal chest abnormalities in COVID-19 patients, but the manual process of diagnosis can be time-intensive, particularly when dealing with a large patient population. Artificial Intelligence (AI), in particular, deep learning (DL) methods like Convolutional Neural Networks (CNNs), has the potential to accelerate the diagnosis of COVID-19 by swiftly identifying abnormalities in medical images [8-10]. Numerous studies have explored machine learning (ML) techniques for diagnosing COVID-19, achieving enhanced accuracy [11]. Nevertheless, this endeavor is not without its challenges. Data availability, especially datasets comprising both X-ray and CT images, remains a constraint. Furthermore, the limited pool of patient data poses difficulties. Pre-trained networks, initially designed for non-medical purposes, encounter challenges when applied to medical images. The absence of crucial patient information such as age and gender further complicate the diagnostic process. Implementing complex deep learning architectures demands substantial hardware resources, presenting another challenge in ML-based diagnosis [12, 13].

Previous research has predominantly employed transfer learning and pre-trained models for predictive purposes. However, the variability in COVID-19 patterns, influenced by factors like geography, age, and co-morbidities, complicates the application of general object recognition knowledge gained from other domains. Consequently, transfer learning may not be the suitable approach for COVID-19 diagnosis. Addressing the nuanced patterns associated with COVID-19 stages requires independent models for accurate detection [14-16]. In light of these challenges, this work presents a ML-based approach for COVID-19 diagnosis, with a focus on improving accuracy. The proposed model operates independently in its training and testing phases, eliminating dependencies on generic knowledge from other domains. Furthermore, our approach considers the diversity of patterns within COVID-19 CT scans, an aspect often underrepresented in existing literature. This paper aims to contribute to the field of COVID-19 diagnosis by offering an independent and robust machine-learning model capable of accurately discerning COVID-19 from non-COVID-19 cases. The proposed methodology for COVID-19 detection using deep learning techniques represents a significant step forward in harnessing the power of AI for medical diagnostics. Here are the key contributions and objectives of the approach:

- **Advanced Segmentation**: Traditional methods often struggle with accurate segmentation, especially in the presence of noise or when lesions exhibit subtle contrasts. The introduction of the TLMFCM segmentation technique presents a novel solution to isolate the region of interest (ROI) with higher accuracy, ensuring that subsequent processes operate on the most relevant data.
- **Effective Feature Extraction:** Leveraging the capabilities of the SSAE, the approach efficiently condenses high-dimensional data from CT scans into meaningful, lower-dimensional representations. This not only enhances the model's efficiency but also improves its ability to capture essential characteristics required for accurate diagnosis.
- **Classification:** The LCS-KELM used as a classifier in the proposed model. By optimizing its parameters through the SOA, the methodology ensures that CT scans are classified with high precision into categories of COVID-19, pneumonia, or non-COVID.
- **Mitigation of Class Imbalance:** Recognizing the inherent challenge of class imbalances in medical datasets, the approach incorporates data augmentation techniques. This ensures that minority classes (like COVID-19 positive cases) are adequately represented, leading to a more balanced and generalized model.
- **Efficient Parameter Optimization:** The SOA, inspired by the behavioral patterns of seagulls, is introduced to fine-tune crucial parameters, ensuring the best predictive outcomes and circumventing issues of local minima.

The primary research objectives of the proposed methodology revolve around elevating the accuracy and efficiency of COVID-19 diagnosis using deep learning techniques on CT scans. Firstly, the research aims to enhance the segmentation precision, especially when faced with noise or subtle lesion contrasts, thus ensuring a more reliable foundation for subsequent analyses. Additionally, by employing the SSAE, the objective is to adeptly transform high-dimensional CT scan data into a meaningful, compact representation, capturing the salient diagnostic features. Another significant objective is to ensure top-tier classification accuracy with the LCS-KELM, especially by optimizing its parameters to cater to the unique challenges posed by medical imaging data. Recognizing the pervasive issue of class imbalance in medical datasets, the research also seeks to bolster the representation of minority classes, thereby enhancing the model's generalization and reliability. Lastly, by seamlessly integrating advanced AI techniques with real-world clinical needs, this research intends to bridge the gap between technological innovation and practical application, paving the way for a collaborative, interdisciplinary approach to improved patient outcomes.

The paper is organized as follows: Section 2 presents work related to the detection of COVID-19. Then, in Section 3, our proposed classification methodology is discussed. Section 4 contains experimental analysis of real COVID-19 data, and the outcomes are discussed. Section 5 contains concluding observations and future directions.

2. RELATED WORKS

COVID-19 pandemic forced researcher from all domain to propose solution for the same. In the following section, we have discussed some prominent domains of work and Table 1 provides a summary of these studies, including their respective advantages and limitations.

2.1 Deep learning approaches for COVID-19 detection in medical images

Minaee et al. [3] presented a diagnostic technique to detect COVID-19 patients from chest radiography images using deep learning models. To address the constraint of limited dataset size, transfer learning was employed to fine-tune four widely recognized pre-trained deep neural networks (ResNet18, ResNet50, SqueezeNet, DenseNet-121) using the training images from the COVID-Xray-5k dataset. Models were evaluated on the remaining 3000 images, assessing sensitivity, specificity, Receiver operating characteristic (ROC) curves, precision-recall curves, average predictions, and confusion matrices. Heat maps highlighting potentially infected lung regions were generated. This study requires further analysis of a larger set of COVID-19 images for more reliable accuracy estimation.

Rasheed et al. [13] proposed a COVID-19 diagnosis technique with ML approaches. The model used a chest X-ray (CXR) image as the input. Two classifiers, logistic regression (LR) and CNN were chosen for their speed and efficiency. PCA was employed to reduce dimensionality. Due to the limited availability of labelled training samples, a data augmentation technique using generative adversarial networks (GAN) was used to increase the training dataset size. The study's use of a small dataset hinders broader result generalization, and further validation on larger, diverse datasets is warranted.

Jain et al. [17] proposed a four-phase approach to address the pressing need for rapid and accurate COVID-19 detection. It begins with data augmentation to expand the available dataset and enhance model generalization. Subsequently, preprocessing techniques are applied to prepare CXR images. The core of the methodology lies in the design of two-stage deep neural network models, which aim to distinguish COVID-19-induced pneumonia from healthy cases, bacterial, and other viral pneumonia cases using X-ray images. The proposed method's effectiveness is rigorously evaluated through both training-validation-testing and 5-fold cross-validation procedures.

Karatzoglou et al. [18] reviewed a Support Vector Machines (SVMs) gained popularity due to their simple yet effective approach, leveraging high-dimensional feature spaces while maintaining computational efficiency. This combination of theoretical elegance and practical performance contributed to their widespread adoption across diverse learning tasks.

Sundaravadivelu and Santhanakrishnan [19] proposed a machine learning based model into medical science led to significant progress, particularly in aiding radiologists with tools for medical image processing. Addressing challenges in breast cancer detection, innovative methods like the Comprising Fuzzy C-Means and Artificial Bee Colony optimization (CABC) algorithm, combining Fuzzy C-means (FCM) clustering and Artificial Bee Colony (ABC) optimization, demonstrated promising results in improving accuracy and grading through rigorous comparative analysis, alongside techniques like Random Forest, further enhancing precision and classification accuracy.

Mulani et al. [20] proposed a system integrating image analysis and machine learning, utilizing CNNs and Decision Trees, achieving a result with precision, recall, and F1-score values of 74.76%, 74.76%, and 74.54% respectively, offering a potential solution to enhance the accuracy and efficiency of skin disease diagnosis.

Zouhal and Denoeux [21] proposed a learning procedure for optimizing parameters in the evidence-theoretic k-nearest neighbor rule, a classification method rooted in the Dempster-Shafer theory. It addresses the unresolved issue of parameter tuning by minimizing an error function, leading to significant enhancements in classification accuracy compared to other methods using the same information, as demonstrated experimentally.

Zhang [22] presented a novel explanation for the impressive classification performance of Naive Bayes, highlighting the role of dependence distribution among attributes. It proposed and proved sufficient and necessary conditions for Naive Bayes optimality, demonstrating how evenly distributed or canceling dependencies contribute to its effectiveness. Furthermore, it investigated Naive Bayes' optimality under the Gaussian distribution, providing evidence that dependencies among attributes may offset each other, shedding light on when Naive Bayes performs well.

Mukherjee et al. [23] proposed a method to address the need for AI-driven tools to detect COVID-19 from radiography and radiology images, including CT scans and CXRs. A CNN-tailored Deep Neural Network (DNN) was engineered to collectively train and test both CT scans and CXRs. The study used both CXR and CT scan images obtained from multiple sources to create balanced datasets for COVID-19 and non-COVID-19 cases. Pre-processing, data augmentation, and deep network model development are the four steps of the suggested technique. By expanding the total length of the dataset, this technique improves model generalization and minimizes the overfitting of the model.

Panwar et al. [24] presented a deep neural network approach to detect COVID-19 with clear visualization using CXR and CT-scan images. A deep transfer learning (DTL) algorithm is introduced to accelerate COVID-19 case detection from radiological images. The study explores the relationship between Pneumonia and COVID-19 in radiological images and uses Grad-CAM for color visualization. Early stopping is employed to mitigate overfitting. The combined form of CNN and DNN is used to train and test image datasets. The approach has been successful in using multiple data in one structure, but, the presence of noise in the image affects the classification accuracy.

2.2 Hybrid and Quantum Approaches with Segmentation

Houssein et al. [25] presented a Hybrid Quantum-Classical Convolutional Neural Network (HQCNN) model for COVID-19 prediction using chest radiography images. The HQCNN model consists of two parts: a quantum part with a quantum Conv layer and a classical part with CNNs. The quantum Conv layer is based on the quantum circuit. The classical part comprises three Conv layers with Rectified Linear Unit (ReLU) activation, two max-pooling layers, and two fully connected layers. The HQCNN model leverages hybrid computation to enhance classical learning but it focuses on binary and multi-class datasets, which may not fully represent the diversity of clinical cases and real-world scenarios.

Abbas et al. [26] classified COVID-19 using Decompose, Transfer, and Compose (DeTraC), which consists of three separate processes: decompose, transfer, and composition. The approach uses Principal Component Analysis (PCA) to reduce the higher-dimensional feature space into a lower one. This helps to reduce memory and improve efficiency. The class decomposition partitions each class into sub-classes. The class decomposition is carried out with the k-means clustering approach. It uses transfer learning to adapt DeTraC for the classification of COVID-19 chest X-ray images. The time complexity of this approach is high.

Das et al. [27] presented an ensemble learning-based COVID-19 detection with CNN. A deep CNN is used to identify the covid-19 patients from the X-ray images. Various CNN models like DenseNet201, Resnet50V2, and Inceptionv3 are trained to observe their predictions. These models are combined with the weighted average ensemble approach to predict the class. The development of a GUI-based application further enhances the practical usability of the model. Also, it achieves high accuracy and sensitivity but its time complexity is high.

Gaur et al. [28] presented a Deep CNN to detect COVID-19 from CXR images. Three pre-trained CNN models were used to evaluate the transfer learning. Transfer learning is used to adapt three specific pre-trained CNN models (EfficientNetB0, VGG16, InceptionV3) for this task. The models are selected since they are suitable for mobile applications. Transfer learning proves beneficial in enhancing model learning capabilities. The time of the training can be reduced through this approach. It discusses misclassification due to image opacity, particularly in the left and right upper lobes and suprahilar areas, which resemble normal X-ray images.

Munusamy et al. [29] created a unique FractalCovNet architecture for both CT-scan image segmentation and CXR image classification. It employs U-Net architecture with Fractal blocks, designed to automate COVID-19 detection. The FractalCovNet model is first trained for CT-scan image segmentation and subsequently fine-tuned for CXR image classification using transfer learning. The percentage of the infection can be determined by this method but the time complexity is high.

Jangam et al. [30] presented a deep learning-based COVID-19 detection strategy with the help of X-ray images. The model is a stacked ensemble consisting of four pre-trained deep learning models such as VGG 19, ResNet 101, DenseNet 169, and Wide Residual Network (WideResNet 50 2). The models are fine-tuned to detect COVID-19. Three best-performing diverse models from the base models are selected to form a weighted averaging-based heterogeneous stacked ensemble.

Transfer learning is employed to leverage pre-trained model weights for faster training and fine-tuning for the COVID-19 detection task. To train the model, five images from X-ray and a chest scan were used for training. The ensemble approach performed better than the basic models and consumes more time.

Table 1. Comparative analysis with existing methods

Author	Year	Methodology	Type of Image	Advantages	Limitations
Minaee et al. [3]	2020	DTL	CXR images	High sensitivity, ROC, and specificity	Not reliable for large-scale data
Rasheed et al. [13]	2021	PCA+ CNN	CXR images	The overfitting problem is reduced	No learning of high abstract features
Jain et al. [17]	2020	alternate diagnostic method	CXR images	It is fast, accurate, reliable, and requires fewer computational requirements	Low sensitivity
Mukherjee et al. [23]	2021	DNN	CXRs and chest CT scan images	It is a lightweight model	Computationally inefficient.
Panwar et al. [24]	2020	DL	Chest CT scan and CXR images	Faster detection	A lot of Misclassification of pneumonia as COVID-19
Houssein et al. [25]	2021	hybrid quantum CNN	CXR	High precision, f1-measure, accuracy, and sensitivity	The size of the images used for training is small
Abbas et al. [26]	2021	DeTraC	CXR	The irregularities in the image are corrected.	Less efficient
Das et al. [27]	2021	deep CNN	CXR	High classification accuracy	The complexity is high
Gaur et al. [28]	2021	Deep CNN	CXR	High accuracy, sensitivity, effective screening and detection measures.	Less opacity leads to misclassification
Munusamy et al. [29]	2021	FractalCovNet	CXR	Cost and time effective	Misclassification of COVID-19
Jangam et al. [30]	2021	stacked ensemble model	chest CT scan and CXRs images	High recall and accuracy	Pre-processing is not efficient
Ni et al. [31]	2020	DL	chest CT images	The reading process is accelerated without affecting the sensitivity	Low specificity

Ni et al. [31] presented a DL based COVID-19 prediction with a chest CT image. This work employs various thresholding methods like adaptive thresholding, binary thresholding, and Otsu thresholding, with preprocessing techniques including noise elimination using filters like Median, Gaussian, and Bilateral. The study achieved improved image quality based on peak signal-to-noise ratio (PSNR) and mean-square error (MSE), with bilateral filtering showing the best results. The combination of Binary and Otsu Thresholding yielded favourable stroke object segmentation. However, further optimization is needed, including edge sharpening and removal of brain skull parts, to enhance segmentation accuracy and reduce false detections of illnesses.

The approach involves segmentation, lesion detection, and location of the lesion. The approach also compared the outcomes of the deep learning with those of radiologists and found the proposed deep learning can achieve better results than the manual approaches.

3. PROPOSED MACHINE LEARNING BASED COVID-19 DETECTION METHODOLOGY

The proposed diagnosis of COVID-19 employs deep learning techniques to find whether the image of the lungs obtained from the CT scan is affected by COVID-19, pneumonia, or non-COVID. The diagnosis is carried out from the images after pre-processing the images, segmentation, feature extraction, and classification steps. The segmentation and feature extraction are carried out with region splitting and a SSAE. The classification is carried out by LCS-KELM. Figure 1 shows the model of the COVID-19 prediction scheme.

The steps involved in the proposed prediction of COVID-19 are as follows.

3.1 Pre-processing

The images in the dataset have different sizes and therefore have varying sizes, so all images were resized with the same dimension. If the images consist of any noise, thresholding is used to remove them [32]. Class imbalance is indeed a prevalent concern in medical imaging datasets, where one class (e.g., COVID-19 positive cases) might be significantly underrepresented compared to other classes (e.g., non-COVID cases). This imbalance can lead to a biased model that performs poorly on the minority class. To mitigate class imbalance issues, data augmentation techniques are applied. This includes:

- **Rotation:** Images are rotated by a certain degree to increase the dataset size.
- **Flipping:** Images can be flipped horizontally or vertically.
- **Zoom:** Images are zoomed in or out to generate slightly different perspectives.
- **Cropping:** Different sections of an image are cropped to create new images.
- **Translation:** Images are shifted either vertically or horizontally.
- **Brightness and Contrast Adjustment:** Modifying the brightness and contrast to simulate different lighting conditions.

In the context of medical images, these augmentations generate new images from the existing dataset, thus increasing the amount of training data, especially for the underrepresented class. This can help in training a more balanced and generalized model.

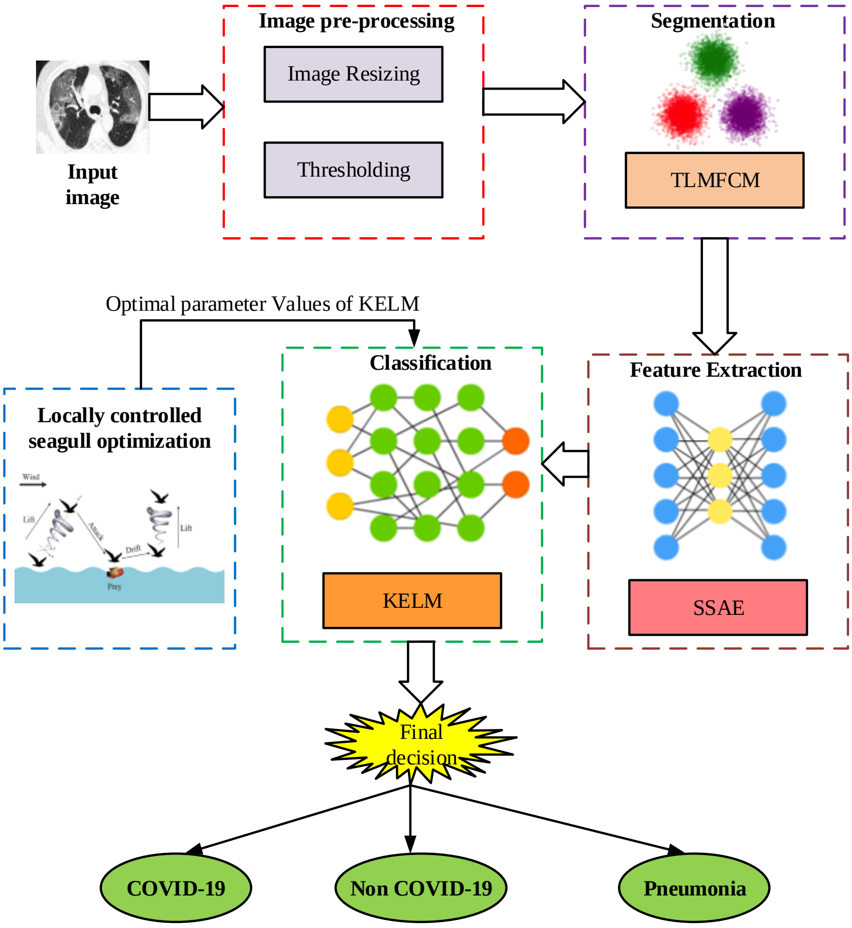


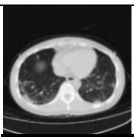
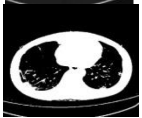
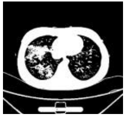
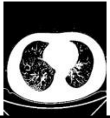
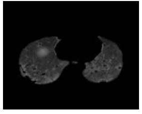
Figure 1. Block diagram of the proposed model

3.2 Segmentation

The segmentation of the image consists of highlighting the ROI. The segmentation process includes the separation of lung lesion regions from the image. The lesions are segmented with the region segmentation technique. The region splitting consists of a split of the region and similar regions are merged. FCM is a traditional clustering approach that provides the capacity to optimally partition data [33, 34]. However, it works effectively only in regions where there is no noise.

FCM ignores the local information that is accessible, as well as the artefacts in the spatial environment. The primary disadvantage of this approach is that it required more iterations to reach a point of convergence. Because of this, the algorithm's time complexity is higher. An efficient clustering algorithm should be able to do the following tasks: (i) Reduce local optima problem, (ii) Effectively measure the similarity and differentiate elements even in the smooth contour area, (iii) The algorithm should be tolerant to noise (iv) Minimise time complexity; clustering techniques often require a long time to converge, which must be minimised. However, the FCM approach may incorrectly segment the regions due to the presence of noisy pixels. Therefore, we present Tolerant Local median FCM, which incorporates features such as local optima minimization, fuzzy clustering, effective clustering in the smooth contour region, and noise tolerance. Table 2 gives the pre-processed and segmented output of the proposed approach. The TLMFCM algorithm addresses the existing FCM's incapacity to deal with noisy data and low precision in deriving clusters of the contour area.

Table 2. Pre-processed and segmented output of the proposed approach

Class	COVID-19	Common Pneumonia	Normal
Input			
Pre-processing			
Segmentation			

In TLMFCM, a data vector is utilised for the operation since the 2-Dimension (2D) data makes it harder. When iteration begins, it estimates the median value of the window selected. This parameter is responsible for overcoming the complexity issues and F_{mmfi} minimises local optima problems. The cluster centre modification procedure is given by Eqs. (3)-(6). Eq. (3) is a common technique of producing a cluster centre, which returns the currently available cluster. However, for faster convergence and better parameter selection, Eq. (5), which is a median adjustment distance is utilised for the adjustment of the final cluster centre by computing the distance between the cluster centre of the previous iteration and the present iteration. This gives Eq. (4) which will be immediately applied for the optimal adjustment of the existing cluster centre. The procedure for TLMFCM is described as follows:

The FCM parameters such as the number of clusters, maximum iteration, convergence criteria, and fuzzy factor are initialized. During the first iteration, a random membership function is built using the available data. Since the 2D data makes the operation harder, the operation is done with a data vector V_{xs}.

The median adjustment parameter is determined using Eq. (1).

$$F_{mmfi} = \sum_{s \in K_s} \frac{Med(Q_{st})}{Med(Q_{st})+1} \left\| V_{xs} - CC_t \right\|^2 \tag{1}$$

where, $Med(Q_{st})$ signifies the median of the chosen window, V_{xs} signifies the data vector, CC_t signifies the cluster centre. A random membership function with $K \times \mathrm{TC}$ is created to start the iterative process as given in Eq. (2).

$$J_{itr} = \sum_{s=1}^{K} \sum_{t=1}^{TC} \left[F_{mmfi} + M_{fun} \left\| V_{xs} - CC_t \right\|^2 \right] \tag{2}$$

where, K and TC signify the maximum number of pixel elements and a number of clusters respectively, and M_{fun} indicate the membership function. The cluster centre that is presently available is determined using Eq. (3).

$$C_{pr\_t} = \frac{\sum_{s=1}^{K} M_{fun} V_{xs}}{\sum_{s=1}^{K} M_{fun}} \tag{3}$$

The distance of median adjustment to correct the final cluster centre is determined using Eq. (4).

$$C_{aj} = \frac{C_{aj\_M}}{\max(P_v)} \tag{4}$$

where, C_{ajM} indicates the median adjustment distance and is found using Eq. (5).

$$C_{aj\_M} = Med\left(\left\| C_{old\_t} - C_{pr\_t} \right\| \right) \tag{5}$$

where, C_{old_t} and C_{pr_t} indicates the old and present cluster centres respectively. The final cluster centre is calculated using Eq. (6).

$$C_t = \left(C_{pr\_t} \right)^{1/C_{aj}} \tag{6}$$

According to the obtained final cluster centre, the membership function M_{itr} is determined or updated sing Eq. (7).

$$M_{itr} = \frac{1}{\sum_{l=1}^{TC} \left(\frac{F_{mmfi} + dst}{F_{mmfi} + dks} \right)^{1/f-1}} \tag{7}$$

where, dst and dks indicates the distance between V_{xs} and CC_t in various forms. The objective function will be determined during each iteration using J_{itr} Eq. (2).

Finally, the difference between J_{itr} and $J_{itr\text{-}1}$ (the preceding iteration's objective function value) is computed to ensure that the convergence requirements are met. If the value of the objective function does not meet the setup requirements, the cycle repeats again until the convergence is reached.

The implementation details of the TLMFCM algorithm are depicted in algorithm 1 as follows:

Window Size: The choice of the window size has implications for the granularity of segmentation. After multiple trials, we settled on a window size of 3×3 for our CT images. This size efficiently captures the local information without being overly sensitive to noise or overly generalizing the features.

Algorithm 1: TLMFCM segmentation

```
Input: Pre-processed training dataset
Output: Segmented image
Initialize the parameters and maximum iteration
for the pre-processed training datasets do
        Calculate the median adjustment parameter $F_{mmfi}$
        Start the iterative process $J_{itr}$
        Determine the present cluster centre $C_{pr\_t}$ using Eq. (3).
        Calculate the distance of median adjustment $C_{aj}$
        Compute the final cluster centre $C_t = (C_{pr\_t})^{1/C_{aj}}$
        Update the membership function $M_{itr}$
        Compute the difference between $J_{itr}$ and $J_{itr-1}$
end forℝ
if the convergence requirements are met
    Cluster the regions
else
    Repeat the process until the termination condition satisfies
end if
```

Number of Clusters: The choice of the number of clusters typically depends on the inherent structure of the image data. For our dataset, we chose three clusters- representing the background, lung tissues, and lesions or anomalies. The decision is based on our understanding of the typical CT images and preliminary experiments.

Parameters: The TLMFCM algorithm relies on a few parameters:

Fuzziness Factor: The algorithm uses a fuzziness factor of 2. This value provides an optimal balance between hard and soft clustering for our dataset.

Convergence Criteria: A critical component for the efficiency of our algorithm is the convergence criteria. We adopted a threshold value of 1×10^{-5} for the change in consecutive objective function values. If the difference between the current and the previous objective function value falls below this threshold, the algorithm is deemed to have converged.

Maximum Iteration: While the convergence criteria usually suffice, it's prudent to have a fallback to prevent indefinite looping. Thus, we have set a maximum iteration count of 1000 for our TLMFCM algorithm.

Initialization: The FCM parameters, such as the number of clusters and convergence criteria, were initially set. During the first iteration, a random membership function was constructed using the available data. This approach ensured randomness in the initial phase, allowing a broad exploration of the solution space.

3.3 Feature extraction

Seven types of features like gray level dependence matrix (GLDM) features, neighbouring Gray-tone difference matrix (NGTDM) features, gray level run length matrix (GLRLM) features, gray level size zone matrix (GLSZM) features, gray level co-occurrence matrix (GLCM) features, and first-order histogram-based features are used for feature extraction. Therefore, a total of 85 features are extracted to classify COVID-19 patients [35]. Once the lesions are segmented, feature extraction is carried out with a stacked sparse auto-encoder.

When a nonlinear function specifies the relationship between independent and dependent features, autoencoders are employed to minimise the dimensionality of data. Autoencoders are a sort of unsupervised artificial neural network used to extract features from data automatically. It is one of the most effective feature extraction methods, and it is utilized in a variety of applications, including voice recognition, human gesture detection, and self-driving cars. In a sparse autoencoder, the Kullback-Leibler divergence (KL) is added to attain sparsity [36].

The features of some datasets have a complicated relationship. As a result, utilizing just one Autoencoder is insufficient. It is possible that a single Auto encoder won't be able to decrease the input features' dimensionality. As a result, stacked autoencoders are employed in these situations. Multiple encoders are stacked above one another in a stacked autoencoder. The SSAE has an input layer, three hidden layers, and an output layer. By stacking numerous SAEs together, the expressive capacity of features retrieved by a sparse autoencoder is enhanced. The small dataset can cause gradient disappearance and overfitting. This can be avoided by employing a stacked sparse auto-encoder. The outcome of the first SAE is connected to the output of the second SAE. In the second SAE, more expressive characteristics will be extracted. This output is connected to the next SAE for improved feature representation and so on. The SSAE consists of an input layer, 3 hidden layers, and an output layer. 200 hidden nodes are present in the first SAE. The features that are extracted by the 1st SAE are represented as given in Eq. (8).

$$p_i^1 = \sigma\left(WT^1 y_i + b^1\right) \tag{8}$$

where, b, WT, σ and y_i are the bias, weight, activation functions, and input respectively. The output (higher level features) of the 1st hidden layer is then given as input to the 2nd SAE that has a hidden layer with 50 nodes. The 2nd layer further encodes in a higher-level representation as given in Eq. (9).

$$p_i^2 = \sigma\left(WT^2 y_i + b^2\right) \tag{9}$$

The output from the 2nd layer is given to the third layer which consists of 10 nodes in the hidden layer which further encodes into higher level representation as given in Eq. (10).

$$p_i^3 = \sigma\left(WT^3 y_i + b^3\right) \tag{10}$$

The SSAE technique extracts many expressive features by stacking several SAEs as given in Figure 2. The high-level features extracted from the SSAE are given as input to the output layer (LCS-KELM classifier) that has 3 nodes.

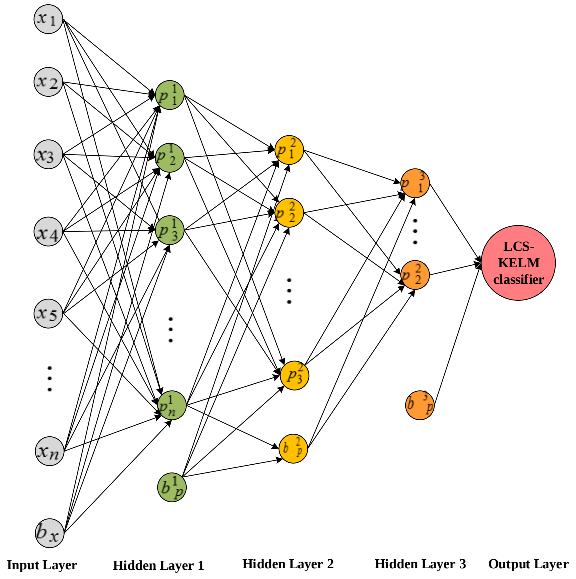


Figure 2. Architecture of SSAE

The implementation details of SSAE is provided below:

Number of Layers and Nodes per Layer: Our SSAE comprises an input layer, three hidden layers, and an output layer. The first hidden layer has 200 nodes. This layer captures primary features from the raw input. The second hidden layer further processes these features and consists of 50 nodes. The third hidden layer, designed to capture even more abstract features, has 10 nodes. The number of nodes in the output layer matches the feature size needed for the LCS-KELM classifier, which is three in our case.

Activation Functions: For our SSAE, we employed the sigmoid activation function for the hidden layers. This choice was based on the sigmoid property of capturing non-linearity's and its effective gradient propagation. For the output layer, we utilized a linear activation function to obtain raw feature values for the subsequent classification stage.

Sparsity Parameters: The KL divergence regularization was added to induce sparsity in the SSAE. The desired average activation value, often denoted as rho, was set at 0.05. This ensures a low average activation for the nodes, encouraging sparsity. The sparsity regularization parameter, typically denoted as beta, was set at 3, ensuring that the sparsity constraint is prioritized in the model.

Optimizer Used: We employed the Adam optimizer for training the SSAE. Adam has shown consistent performance in handling deep architectures like autoencoders. It dynamically adjusts learning rates for each parameter, allowing for faster convergence and efficient handling of sparse data. The initial learning rate for Adam was set at 0.001, and default values were used for other parameters (beta1=0.9, beta2=0.999).

3.4 Classification

In this phase, the LCS-KELM is used to classify whether the CT images belong to COVID-19, pneumonia, or non-COVID-19. The extreme learning machine (ELM) concept of single hidden layer Feed Forward Neural Networks (FFNN) was introduced in 2005 [37]. The LCS-KELM classifier as depicted in algorithm 2 predicts the presence of COVID-19 by using the features that are learned from the third hidden layer of SSAE. The number of nodes in the LCS-KELM is taken as three since there are three groups to be classified.

The kernel function's width η and the regularization coefficient C_{reg} and are two essential factors in KELM. Since these two parameters have no theoretical foundation, tuning the model is required to find the best combination of the parameters. Therefore, the parameters η and C_{reg} are optimized by using a LCS optimization algorithm.

3.4.1 Seagull optimization algorithm

Seagulls are a type of coastal bird that has been around for around thirty million years and can be found in nearly every area of the world. Even though fish is the prime source of food for seagulls, they also eat a variety of things, including moles, earthworms, amphibians, reptiles, and insects. Seagulls are intelligent birds with a lifespan of 10-15 years.

Seagulls usually dwell in swarms. They live in big communities and interact with fellow group members using various voices. They steal food from other animals, birds, and sometimes people, which is one of their weirdest habits. They also employ various methods of hunting. For instance, they use bread crumbs or make a rain shower noise with their feet to attract fish. Another feature is their migration behaviour.

Seagull migration refers to the movement of seagulls northward in the spring and southward in the fall, from the ground to the heights, or from coast to coast, to avoid harsh winter weather and find the most plentiful food sources. This process is believed to be a regular activity of seagulls moving from one location to another in search of a diverse variety of abundant food sources to maintain good energy levels [38].

The migration begins with a swarm of seagulls. Their original locations are distinct from one another to avoid collisions.

- They use their swarm experience to migrate towards the best survival direction.
- Seagulls generally attack migratory birds in a spiral shape behaviour over the sea.

(i) Initialization

The population of the seagull is initialized and then the maximum number of iterations is set. The initial position of the seagull is taken as the initial parameters (random values of η and C_{reg}) and the prey is taken as the optimal values of η and C_{reg}. The fitness of each possible solution (η and C_{reg}) is evaluated and the solution with minimum fitness value is updated regularly. Finally, the solution with the least fitness value (output) is taken as the optimal parameters values of η and C_{reg}. Optimizing these parameters helps to accurately classify the images fed to the KELM classifier.

(ii) Fitness evaluation

The prime objective function is to minimize the fitness function. The fitness of the solution is evaluated by using Eq. (11).

$$fitness = \sqrt{\frac{1}{T_{sam}}\sum_{i=1}^{T_{sam}} Vact_i - Vpre_i} \tag{11}$$

where, $Vact_i$ and $Vpre_i$ indicate the actual and predicted values respectively, and T_{sam} indicate the number of training samples.

(iii) Migration of seagulls (exploration)

The algorithm simulates how a flock of seagulls migrates from one location to another during migration. A seagull must meet three conditions during this phase:

(a) Avoidance of collisions

A model is specified by extra variable A for upgrading the new location of the seagull in order to minimize collisions with neighbouring seagulls (search agent).

The search agent's location without colliding with another search agent is found using Eq. (12).

$$\vec{L}_c = B\times \vec{L}_{c(m)}, \qquad m = 0,1,\ldots,\max(m) \tag{12}$$

where, m denotes the current iteration, $\vec{L}_c$ denotes the current location of the search agent, and B denotes the search agent's movement behaviour in a particular search space which is determined using Eq. (13).

$$B = f_d - \left(m\times\left(\frac{f_d}{\max(m)}\right)\right) \tag{13}$$

where, f_d indicates the variable B's frequency control in the interval $[0, f_d]$.

(b) Movement towards the best solution

The search agents change or renew their location depending on the best result using Eq. (14).

$$\vec{E}_c = A\times\left(\vec{L}_{bt}(m) - \vec{L}_c(m)\right) \tag{14}$$

where, $\vec{E}_c$ signifies the candidate's location, $\vec{L}_{bt}(m)$ in relation to the best-fit candidate $\vec{L}_c(m)$, A signifies a random number that determines whether to pursue exploration or exploitation, which is computed using Eq. (15).

$$A = 2\times B^2\times Rd \tag{15}$$

where, Rd indicates a random value in the range [0, 1]

(c) Movement towards the best search agent

Lastly, the search agent can update its location with respect to the best solution using Eq. (16).

$$\vec{D}_c = \left|\vec{L}_c + \vec{E}_c\right| \tag{16}$$

where, $\vec{D}_c$ denotes the search agent's distance from the best solution.

(iv) Attacking the prey (exploitation)

The goal of the exploitation is to make use of the search process's experience and history to get the best solution. During migration, seagulls may adjust their assault angle and pace. They use their weight and wings to keep their altitude. The spiral motion happens in the air when hunting the prey. The following Eq. (17) to Eq. (19) is a description of this behaviour in the x, y, and z planes.

$$x' = k\times\cos(r) \tag{17}$$

$$y' = k\times\sin(r) \tag{18}$$

$$z' = k\times r \tag{19}$$

where, r signifies a random value in the range $[0 \le r \le 2\pi]$ and k signifies the spiral turn's radius, which is determined using Eq. (20).

$$k = e^{\alpha r} \times \beta \tag{20}$$

where, α and β are the constants that define the spiral shape. The seagull's new location is updated using the Eq. (21).

$$\vec{L}_{c}(m) = \left(\vec{D}_{c} \times z' \times y' \times x'\right) + \vec{L}_{bt}(m) \tag{21}$$

where, the best solution is saved in $\vec{L}_c(m)$.

3.4.2 Locally controlled SOA

Lévy flight (LF) has been utilised to improve the suggested SOA in this section. This technique is used to alleviate the premature convergence problem, which is the SOA's major drawback. LF is a random walk technique for effective local search control [39].

The balance between exploration and exploitation is critical in practically all meta-heuristic algorithms to ensure excellent performance. A high level of variety in the population should be maintained so that exploration and exploitation can be represented in the evolving population. If diversity is insufficient, local exploitation may result in early convergence, and therefore the chance to reach the global optimum may be lost. LF has been shown in studies to increase the efficiency of the resource-seeking process in unpredictable circumstances. Using LF, the global search space may be explored more efficiently. A LF approach is combined with the SOA algorithm to balance exploration and exploitation.

LF is defined using Eq. (22).

$$Lv(q) \approx q^{-1-\sigma} \tag{22}$$

where, q signifies the step size and $0<\sigma<2$.

$$q = \frac{F}{|G|^{1/\sigma}} \tag{23}$$

where, $F \sim N(0, \tau^2), G \sim N(0,1)$

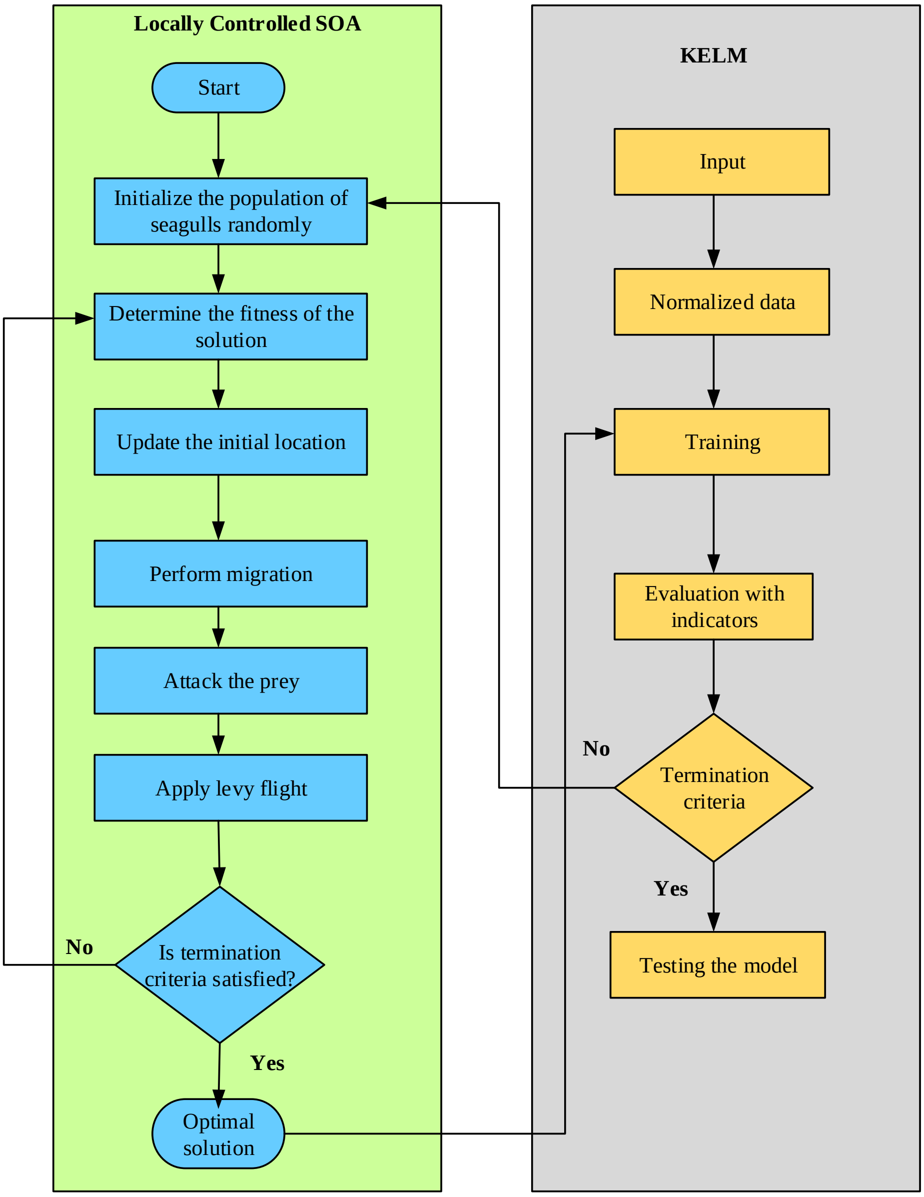


Figure 3. Flowchart for LCS-KELM

$$\tau^2 = \left\{ \frac{\Gamma(1+\sigma)}{\sigma\Gamma((1+\sigma)/2)} \frac{\sin\left(\pi\sigma/2\right)}{2^{(1+\sigma)/2}} \right\}^{2/\sigma} \tag{24}$$

where, $\Gamma(.)$ denotes the gamma function.

The location of the search agent is updated using Eq. (25).

$$\vec{D}_{cl} = \vec{D}_c + \left|\vec{L}_c + \vec{E}_c\right| + Lv(q) \tag{25}$$

where, $\vec{D}_c$ denotes the previous location of search agent, $\vec{L}_c$ denotes the current location of the search agent, $\vec{E}_c$ signifies the candidate's location, and $Lv(q)$ denotes LF.

$$\vec{D}_{cl} = \begin{cases} \vec{D}_{cl} & F(\vec{D}_{cl}) > F(\vec{D}_c) \\ \vec{D}_c & otherwise \end{cases} \tag{26}$$

where, $\vec{D}_c$ denotes the search agent's distance from the best solution, $\vec{D}_{cl}$ denotes the updated location of the search agent. The seagull stops hunting for prey i.e., the output is obtained (optimal η and C_{reg} parameter values of KELM) when the maximum number of iterations is reached or when the best values of the KELM parameters are found. Figure 3 gives the flowchart for the LCS-KELM.

Algorithm 2: LCS-KELM-based classification

```
Input: Extracted features
Output: Classified output: COVID-19, normal, pneumonia
Initialize the population, maximum iteration
while current iteration<maximum iteration do
    Determine the fitness of the solution
    Update the initial location
    Migrate the seagull from one location to another
    Attack the prey (solution)
    Apply levy flight $Lv(q) \approx q^{-1-\sigma}$
end while
Return $\eta$ and $C_{reg}$
if termination condition satisfies
    Provide the values of the parameters $\eta$ and $C_{reg}$ to KELM
else
    Repeat the process until current iteration>=maximum iteration
end if
```

4. SIMULATION SETUP AND COMPARATIVE ANALYSIS

The simulation is conducted in python using the COVID-19 dataset collected from the reference [40]. The experiments utilized a machine running on Windows 10 operating system, powered by an Intel Core i9-7900X CPU with a clock speed of 3.30 GHz. The system also featured an NVIDIA GeForce GTX 1080 Ti graphics processing unit and had 32.0 GB RAM. The following segments delve deeper into the details of the dataset, assessment criteria, and the resulting experimental findings.

4.1 Model components

The primary components of the diagnostic model include the SSAE and the LCS-KELM. The SSAE, a specialized deep neural network, is constructed with an input layer, three progressively smaller hidden layers, and an output layer. Starting with 200 nodes in the first hidden layer, the architecture ensures a hierarchy of feature abstraction as the number of nodes decreases in subsequent layers. The SSAE's main purpose is to condense the high-dimensional CT scan images into a compact yet informative representation, highlighting the crucial diagnostic characteristics. On the other hand, the LCS-KELM, designed for swift training and superior generalization, consists of a single hidden layer. Leveraging the feature set derived from the SSAE, the LCS-KELM precisely classifies the CT scan images into one of three categories: COVID-19, Normal, or Pneumonia. Together, these components create an efficient and accurate diagnostic system.

4.2 Training process

The training process harnesses features extracted by the SSAE, employing the LCS-KELM classifier optimized by the SOA. Through 10-fold cross-validation, each data point is methodically utilized for both training and testing, ensuring robust model accuracy.

(a) Dataset Description and Source:

The dataset used for the simulations is termed as the "COVID-19 dataset", which was collected from reference [40]. The dataset comprises images of lungs obtained from CT scans. These images are primarily grouped into three categories:

- COVID-19
- Common Pneumonia
- Normal (indicating no presence of the COVID-19 or Pneumonia).

(b) Characteristics and Size:

The dataset's images vary in their initial dimensions, prompting the need for pre-processing. For consistent analysis, all the images were resized to a standard size, 512x512. The dataset's composition consists of 2035 images for the COVID-19 class, 2119 images for the normal class, and 3390 images for the pneumonia class, leading to a total of 7544 images.

(c) Feature Extraction using SSAE:

In the feature extraction phase, the post-segmented images from the dataset are processed using a SSAE. SSAE, a variant of deep neural networks, serves as an effective tool for distilling high-dimensional data from images into a compact, lower-dimensional format. Structurally, the SSAE is composed of an input layer, followed by three hidden layers, and culminating in an output layer. Intriguingly, each hidden layer is characterized by a varying number of nodes. For instance, the inaugural hidden layer boasts 200 nodes. As the data progresses through these layers, it undergoes a transformation, with each layer encapsulating the features into increasingly abstracted and succinct representations.

(d) Parameter Optimization using SOA:

To ensure that the KELM operates at peak efficiency, it's crucial to fine-tune its parameters meticulously. This optimization process leverages the SOA to pinpoint the ideal values for the kernel function's width and the regularization coefficient inherent to the KELM. Drawing inspiration from natural behaviors exhibited by seagulls, such as their distinct patterns of migration and hunting, the SOA meticulously adjusts these parameters to maximize the model's predictive capabilities. Further enhancing the algorithm's efficiency and ensuring it avoids the pitfalls of local minima, the inclusion of LF into the SOA ensures a more comprehensive and effective search for the optimal parameter configuration.

(e) Model Training using KELM with Optimized Parameters and 10-cross validation:

Utilizing the refined features procured from the SSAE, the LCS-KELM classifier embarks on its training journey. The core objective of LCS-KELM is to discern whether a CT scan delineates a COVID-19 infection, pneumonia, or a non-COVID instance. Essential to its efficacy is its employment of the optimized parameters ascertained by the SOA, ensuring the pinnacle of predictive accuracy. To bolster the model's credibility and robustness, a 10-fold cross-validation technique is applied. This meticulous approach involves partitioning the entire dataset into ten segments. Over ten iterations, nine of these segments serve as the training set while the remaining segment functions as the test set. This cyclical methodology guarantees that each data point is utilized for both training and testing, reinforcing the model's reliability and mitigating any potential for overfitting. The Table 3 specifies the parameters used in the proposed methodology.

The experiments utilized 10-fold cross-validation as given in Figure 4 to validate the suggested model. Each instance of the dataset is exposed to 80% training and 20% testing at least once by cross-validation. As a result, biased outlier modelling may be avoided.

By employing this rigorous training process and optimizing the parameters with real-world-inspired algorithms like the Seagull Optimization, the proposed methodology showcases improved accuracy, precision, recall, and other metrics over existing methods.

Table 3. Parameters table

Parameter	Description
Dataset source [40]	CT Scan images
Dataset categories	Covid-19, Pneumonia, normal
Image resize dimension	512×512
Noise removal	Thresholding [32]
SSAE layers (Input, Hiddenx3, Output)	5 layers
Hidden nodes in SSAE	200 (1st layer), 50 (2nd layer), 10 (3rd layer)
Optimization algorithm	SOA with Levy flight
KELM parameters optimized	Kernel functions width and regularization coefficient
Validation method	10-fold cross validation
Maximum iterations for SOA	100
Population of Seagulls	50
Dimension for SOA	2

4.3 Evaluation metrics

Metrics such as Area under the ROC curve (AUC), F1 score, Specificity, Recall or Sensitivity, Precision, Accuracy, and run-time are computed in each fold and the obtained results are compared with the existing COVID-19 detection techniques.

(i) Accuracy

The most prevalent criterion for evaluation is accuracy. The greater the classifier's performance, the higher the accuracy. It is computed using Eq. (27).

$$Acc = \frac{S_{TP} + S_{TN}}{S_{FP} + S_{FN} + S_{TP} + S_{TN}} \quad (27)$$

where, S_{TP} and S_{TN} denote True Positive and True Negative respectively, whereas S_{FP} and S_{FN} denote False Positive and False Negative respectively.

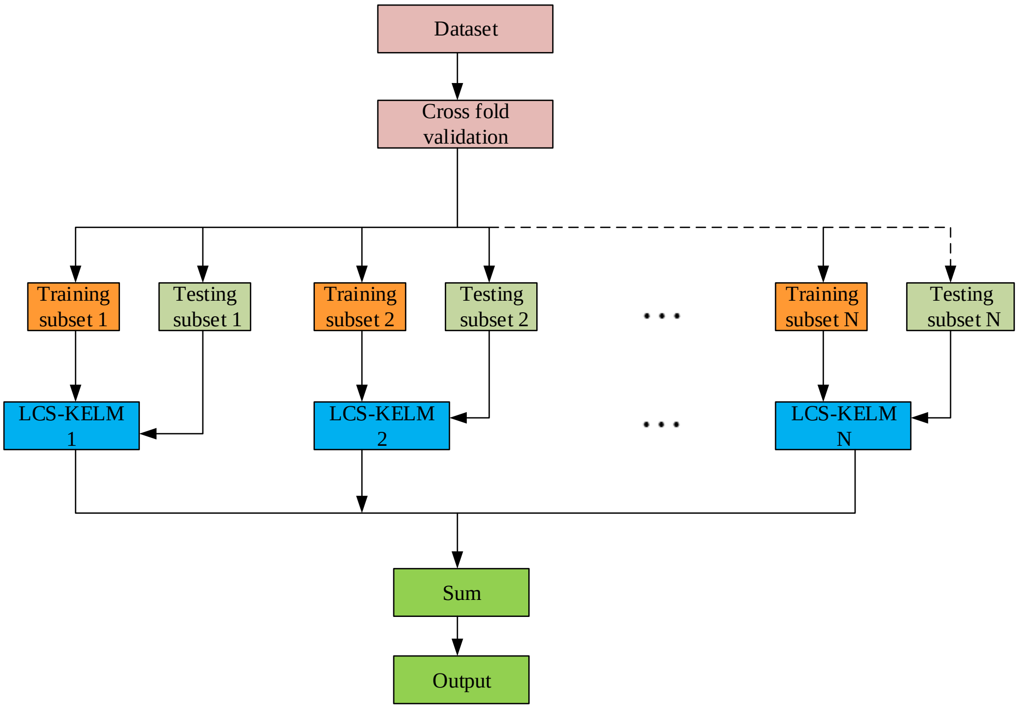


Figure 4. Cross-fold validation

(ii) Recall

The fraction of positive instances in the dataset that is anticipated to be positive is known as recall. It is computed using Eq. (28).

$$rcl = \left(\frac{S_{TP}}{S_{FN} + S_{TP}} \right) \tag{28}$$

(iii) Specificity

The ability of a classifier to detect negative samples is measured by specificity. It is computed using Eq. (29).

$$Spe = \frac{S_{TN}}{S_{FP} + S_{TN}} \tag{29}$$

(iv) Precision

Precision refers to the percentage of positive forecasts that fall into the positive category. It is computed using Eq. (30).

$$pr = \left(\frac{S_{TP}}{S_{FP} + S_{TP}} \right) \tag{30}$$

(v) F1-score

The weighted harmonic average of precision and recall is the F-score and is determined using Eq. (31).

$$F_1 = 2\left(\frac{S_{pr}}{S_{pr} + S_{rcl}} \right) \tag{31}$$

where, S_{rcl} and S_{pr} are the recall and precision scores.

(vi) ROC

The false positive rate (FPR) is plotted against the true positive rate (TPR) to create a ROC curve (FPR). It is determined by using Eq. (32) and Eq. (33) respectively.

$$FPR = \frac{S_{FP}}{S_{FP} + S_{TN}} \tag{32}$$

$$TPR = \frac{S_{TP}}{S_{FN} + S_{TP}} \tag{33}$$

4.4 Qualitative analysis

The qualitative prowess of the proposed methodology is underpinned by several intricate components and strategies, each addressing specific challenges inherent in medical imaging. Firstly, the TLMFCM stands out in its robustness against noisy data, a frequent challenge in CT scans. By emphasizing local median values, it ensures accurate segmentation of lung lesions, even amidst data imperfections. Furthermore, the methodology's adoption of the SSAE provides a sophisticated mechanism for delving deep into high-dimensional data, transforming it into compact representations that retain essential diagnostic characteristics. This becomes pivotal, especially when the relationship between features can be complex and nonlinear, as is often the case with medical images. Moving to classification, the harmonious integration of the LCS-KELM classifier with the SOA ensures meticulous fine-tuning of kernel parameters, drawing inspiration from seagull behaviors for efficient parameter optimization. The inclusion of LF further refines this, striking a balance between exploring new solution areas and honing in on the best-found solutions, thus preventing premature convergence. Collectively, these attributes render the proposed methodology as a qualitatively superior approach, promising enhanced accuracy in the segmentation and classification of COVID-19 from CT scans, a crucial requirement in today's medical landscape.

4.5 Quantitative results

The proposed LCS-KELM classifier is compared with existing techniques such as PCA+CNN [13], DNN [23], HQCNN [25], Deep CNN [26], and Stacked ensemble model [27]. Figure 5 gives the accuracy scores obtained for the proposed LCS-KELM classifier and existing ones under 5-fold and 10-fold validation. Under 5-foldcross-validation, the classification accuracy for PCA+CNN [13], DNN [23], HQCNN [25], Deep CNN [26], and Stacked ensemble model [27] and the proposed LCS-KELM classifier are 74%, 82%, 85%, 88%, 92%, and 97% respectively. This shows that the proposed classifier has higher accuracy than the existing ones. Similarly, when the performance is tested under 10-fold validation, the proposed LCS-KELM classifier shows better classification accuracy of 95.2%. The parameters in the KELM are optimally chosen by the LCS optimization. This LCS prevents premature convergence and increases classification accuracy.

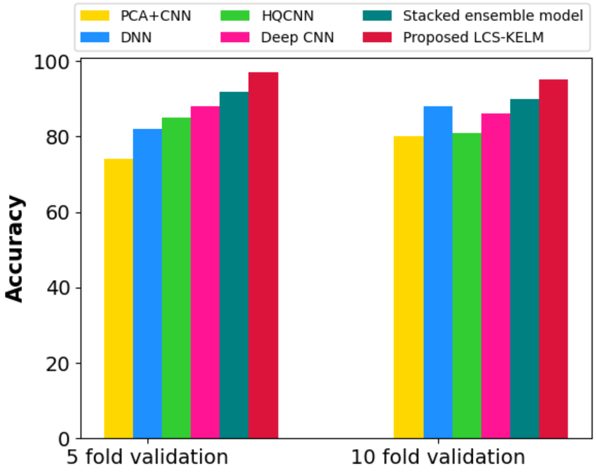


Figure 5. Accuracy analysis

Existing techniques in the realm of COVID-19 diagnosis through imaging have shown promise but come with their share of limitations. PCA+CNN [13], for instance, relies on PCA - a linear dimensionality reduction method that might miss complex non-linear relationships, potentially leading to information loss. DNNs [23], while powerful, grapple with overfitting issues, especially when training data is sparse, and demand significant computational resources. HQCNN offers an avant-garde approach, but its complexity and nascent stage mean its practicality for widespread deployment is questionable. Similarly, the Deep CNN architecture, with its depth, can also succumb to overfitting and demands meticulous tuning. Lastly, Stacked Ensemble Models, though aiming for robustness through model diversity, can be computationally expensive and performance can waver if the base models aren't diverse enough or produce correlated errors.

In contrast, the proposed LCS-KELM approach is crafted to address these challenges. Emphasizing on optimized feature extraction, the methodology harnesses the SSAE's capability to capture essential data characteristics pivotal for precise diagnosis. This nuanced focus on feature extraction ensures crucial data elements are preserved and leveraged effectively. Moreover, the LCS-KELM is renowned for its swift training, enabling rapid model updates and deployments, which is a crucial aspect in the dynamic field of medical diagnostics. The integration of the SOA with the classifier ensures optimal parameter settings, enhancing prediction accuracy. Further, the model's design inherently handles class imbalances and, by leveraging LF, achieves a harmonized balance between exploration and exploitation, reducing the chances of overfitting and boosting generalization. Thus, the proposed model, with its unique synthesis of techniques, seeks to provide an advanced, efficient, and robust solution for diagnosing COVID-19 using CT scan imagery.

The recall score for proposed and existing ones is shown in Figure 6. The recall scores acquired for PCA+CNN [13], DNN [23], HQCNN [25], Deep CNN [26], and Stacked ensemble model [27], and the proposed LCS-KELM classifier is74%, 72%, 85%, 81%, 88%, and 91% respectively. The suggested LCS-KELM classification technique produces superior outcomes based on the analysis with the existing techniques. This aids in properly identifying the patients affected by COVID-19. A single Autoencoder won't be able to minimise the input features' dimensionality. The proposed model employs a stacked model rather than a single model for extracting the features. This improves the recall or Sensitivity score compared to other schemes.

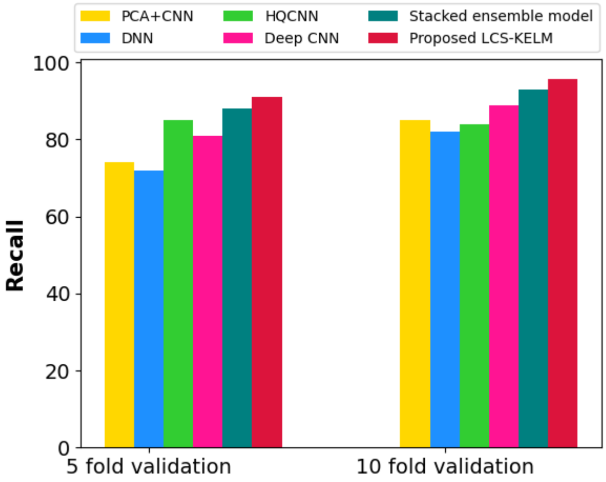


Figure 6. Recall analysis

The specificity obtained for the suggested approach is depicted in Figure 7. When testing the performance of the proposed LCS-KELM classification method, it is observed that the proposed LCS-KELM classifier offers higher specificity (97%) than the existing techniques such as PCA+CNN [13] (65%), DNN [23] (73%), HQCNN [25] (77%), Deep CNN [26] (80%), and Stacked ensemble model [27] (81%). One of the factors that improve the specificity of the proposed method is due to the elimination of noise in the image by the binary thresholding method. This helps to accurately segment and classify the images.

The precision analysis for the suggested approach is depicted in Figure 8. When testing the proposed LCS-KELM classification method in the COVID-19 dataset, it is observed that the proposed LCS-KELM classifier offers higher precision (93%) than the existing techniques such as PCA+CNN [13] (68%), DNN [23] (75%), HQCNN [25] (74%), Deep CNN [26] (85%), and Stacked ensemble model [27] (91%). In the proposed methodology, the TLMFCM segmentation technique clusters the data by using the fuzzy methodology and also reduces the local optima problem. Compared to the existing techniques, the TLMFCM segmentation technique is able to differentiate elements in smooth contour areas. This increases the precision during classification.

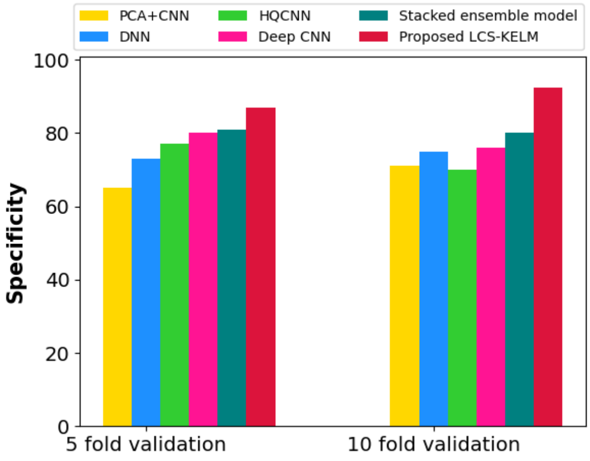


Figure 7. Specificity analysis

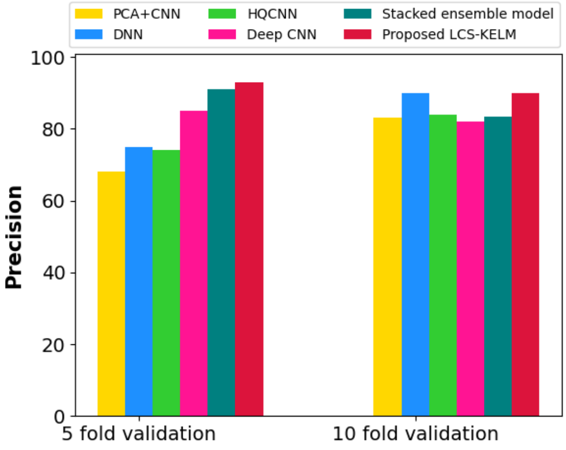


Figure 8. Precision analysis

The performance analysis is shown in Figure 9 in terms of the F1-score. The proposed LCS-KELM classifier outperforms the current systems while computing F-score. The F1-score for PCA+CNN [13], DNN [23], HQCNN [25], Deep CNN [26], and Stacked ensemble model [27] and the proposed LCS-KELM classifier are 62%, 70%, 75%, 71%, 84%, and 91% respectively.

From Figure 10, it is observed that the proposed LCS-KELM classifier has a higher AUC than the other models such as PCA+CNN [13], DNN [23], HQCNN [5], Deep CNN [26], and Stacked ensemble model [27]. The proposed LCS-KELM model has the highest AUC of 0.988 while DNN has the lowest AUC of 0.959. Table 4 and Table 5 show the numerical results obtained for the proposed and the existing approaches for 5-fold and 10-fold validation respectively.

Table 4. Comparative analysis of overall performance with existing techniques for 5-fold validation

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-score	Specificity (%)
PCA+CNN [13]	74	68	74	62	65
DNN [23]	82	75	72	70	73
HQCNN [25]	85	74	85	75	77
Deep CNN [26]	88	85	81	71	80
Stacked ensemble model [27]	92	91	88	84	81
Proposed LCS-KELM classifier	97	93	91	91	87

Table 5. Comparative analysis of overall performance with existing techniques for 10-fold validation

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-score	Specificity (%)
PCA+CNN [13]	80	83	85	70	71
DNN [23]	88	90	82	71	75
HQCNN [25]	81	84	84	80	70
Deep CNN [26]	86	82	89	85	76
Stacked ensemble model [27]	90	83.5	93	82	80
Proposed LCS-KELM classifier	95.2	90	95.8	95	92.5

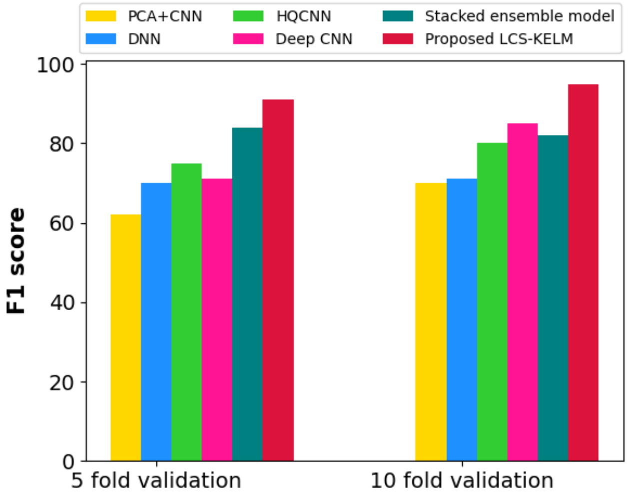


Figure 9. F1-score analysis

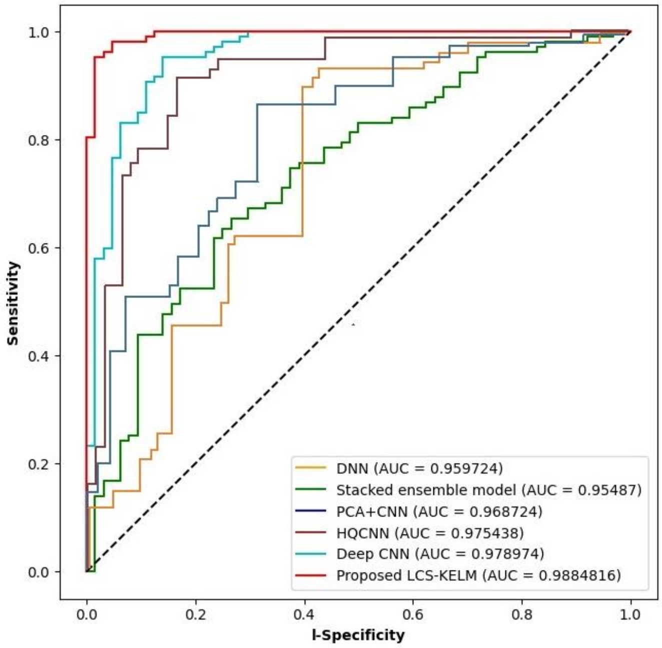


Figure 10. ROC analysis

Figures 11-13 provide a comprehensive comparative analysis of segmentation in terms of accuracy, sensitivity and specificity in the context of COVID-19 diagnosis from medical images. Figure 11 illustrates accuracy percentages, showcasing the proposed LCS-KELM classifier's remarkable precision, surpassing other methods like PCA+CNN, DNN, HQCNN, Deep CNN, and Stacked Ensemble Model. The second graph, focusing on sensitivity percentages, highlights the superior ability of the LCS-KELM classifier to accurately detect COVID-19 cases. The third graph emphasizes specificity percentages, revealing the classifier's excellence in correctly identifying non-COVID-19 cases. Together, these graphs provide a clear visual representation of the outstanding segmentation performance of the proposed method in terms of accuracy, sensitivity, and specificity, making it a robust tool for precise COVID-19 diagnosis from medical images.

Table 6 provides a comprehensive comparison of segmentation performance between the proposed LCS-KELM classifier and existing techniques, utilizing both 5-fold and 10-fold cross-validation methods. The segmentation performance is evaluated based on key metrics such as accuracy, sensitivity, and specificity.

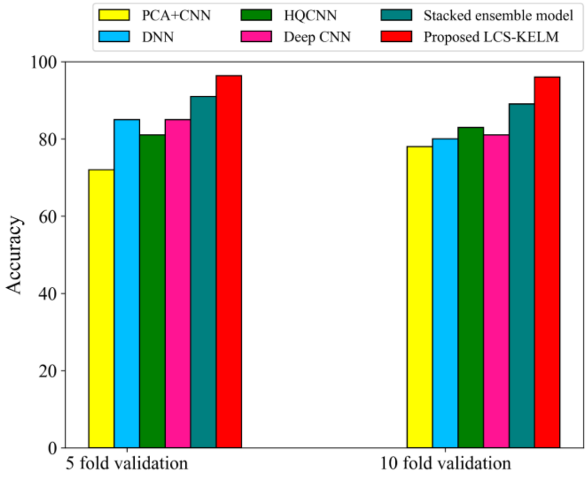


Figure 11. Segmentation accuracy comparison

Figures 14-16 offer an exhaustive comparative analysis of feature extraction performance in terms of accuracy, sensitivity, and specificity using medical images. Figure 14 presents accuracy percentages, prominently showcasing the superior precision achieved by the proposed method in extracting features. It outperforms established methods like

PCA+CNN, DNN, HQCNN, Deep CNN, and Stacked Ensemble Model. Figure 15, dedicated to sensitivity percentages, emphasizes the LCS-KELM classifier's exceptional ability to accurately detect COVID-19 cases during the feature extraction process. Lastly, Figure 16 underscores specificity percentages, in extracting features. These figures collectively depict the remarkable feature extraction performance of the proposed method, demonstrating its superiority in terms of accuracy, sensitivity, and specificity.

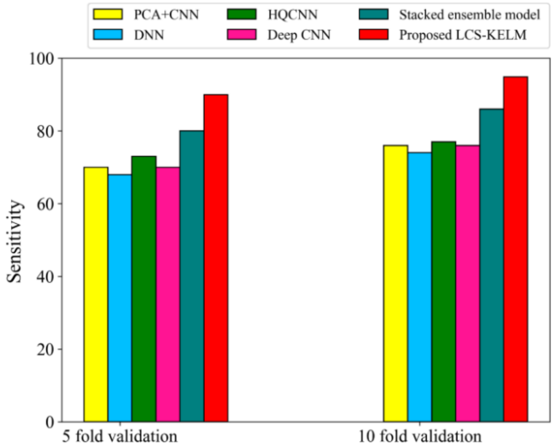


Figure 12. Segmentation sensitivity comparison

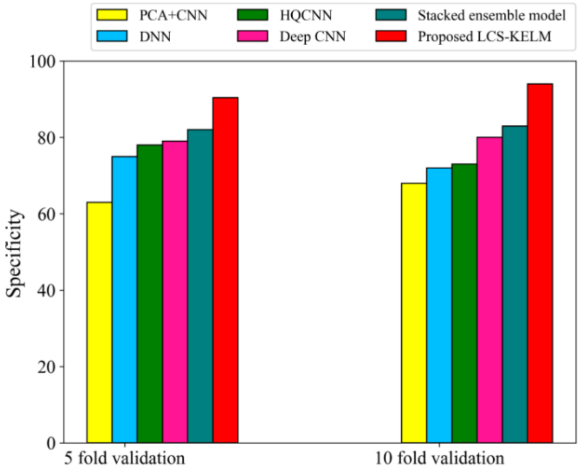


Figure 13. Segmentation specificity comparison

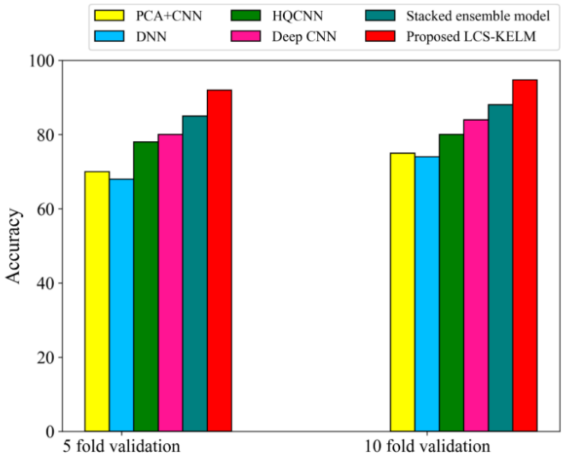


Figure 14. Feature extraction accuracy comparison

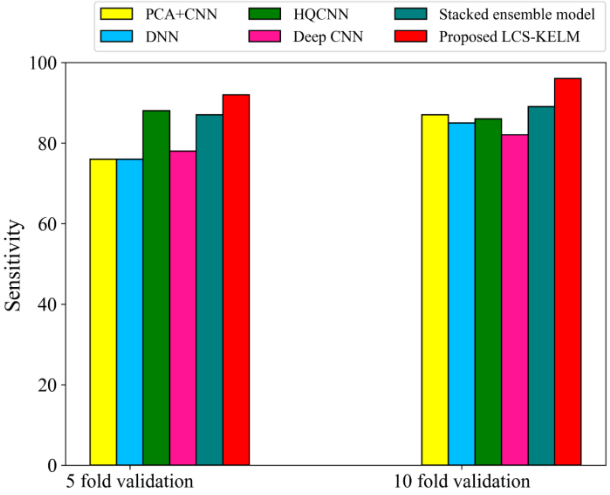


Figure 15. Feature extraction sensitivity comparison

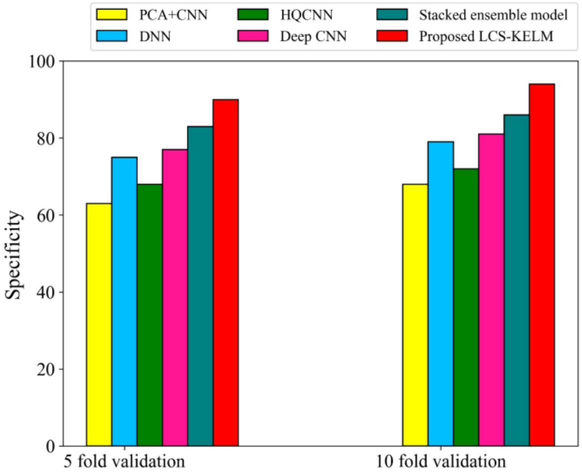


Figure 16. Feature extraction specificity comparison

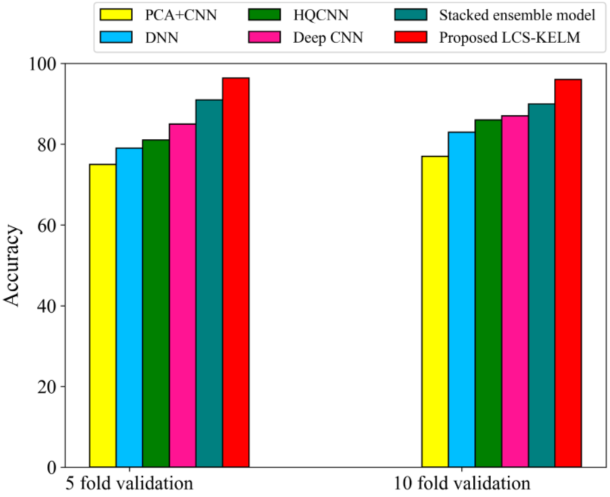


Figure 17. Classification accuracy comparison

Table 7 provides an extensive comparative analysis of feature extraction performance between the proposed LCS-KELM classifier and existing techniques, using both 5-fold and 10-fold cross-validation methods. This evaluation is based on key metrics, including accuracy, sensitivity, and specificity. For the proposed LCS-KELM classifier in the 5-fold cross-validation scenario, the accuracy is 96.4%, sensitivity is 93%, and specificity is 92%. In the 10-fold cross-validation scenario, it achieves an accuracy of 96%, sensitivity of 95.7%, and specificity of 94%. These outstanding numerical results further validate the effectiveness of the proposed method in feature extraction for precise COVID-19 diagnosis from medical images.

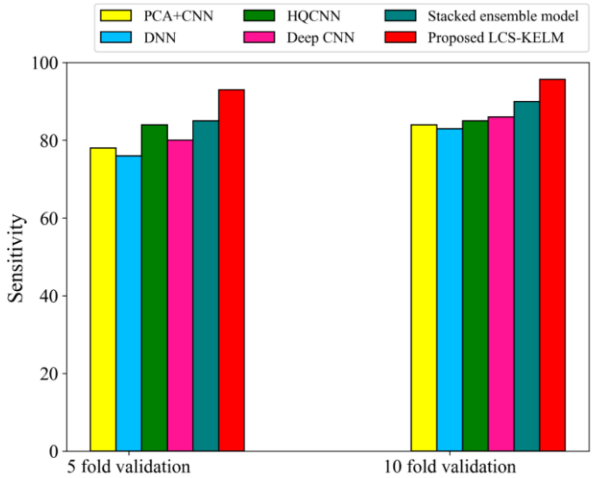


Figure 18. Classification sensitivity comparison

Figures 17-19 offer a comprehensive comparative analysis of classification performance in the context of COVID-19 diagnosis from medical images, focusing on accuracy, sensitivity, and specificity. Figure 17 illustrates accuracy percentages, highlighting the exceptional precision achieved by the proposed LCS-KELM classifier compared to established methods such as PCA+CNN, DNN, HQCNN, Deep CNN, and Stacked Ensemble Model. Figure 18 delves into sensitivity percentages, emphasizing the superior ability of the LCS-KELM classifier to accurately detect COVID-19 cases during the classification process. Finally, Figure 19 underscores specificity percentages, revealing the classifier's excellence in correctly identifying non-COVID-19 cases. These figures collectively demonstrate the outstanding classification performance of the proposed method, showcasing its superiority in terms of accuracy, sensitivity, and specificity.

Table 8 provides a comprehensive comparative analysis of classification performance between the proposed LCS-KELM classifier and existing techniques, utilizing both 5-fold and 10-fold cross-validation methods. This evaluation considers key metrics, including accuracy, sensitivity, and specificity. For the proposed LCS-KELM classifier in the 5-fold cross-validation scenario, the accuracy is 96.4%, sensitivity is 93%, and specificity is 92%. In the 10-fold cross-validation scenario, it achieves an accuracy of 96%, sensitivity of 95.7%, and specificity of 94%. These numerical results underscore the effectiveness of the proposed method in classification for precise COVID-19 diagnosis from medical images, further validating its robustness.

4.6 Comparative analysis with machine learning models

In the study, several baseline ML techniques, including Support vector machine (SVM) [18], Random Forest [19], Decision Tree [20], K-Nearest Neighbors (KNN) [21], and Naïve Bayes [22], were employed to benchmark and draw comparisons with the performance of the newly proposed LCS-KELM classifier in detecting COVID-19. According to the visual representation in Figure 20, the LCS-KELM classifier demonstrated exceptional efficacy, achieving an overall classification accuracy surpassing 96.4%. This is a remarkable accomplishment when considering the variability and complexity of diagnosing conditions like COVID-19. Furthermore, upon analyzing the performances delineated in Figure 20, it becomes evident that LCS-KELM stands out as the top-performing classifier. In contrast, the Decision Tree emerges as the least effective among the evaluated classifiers. The subsequent rankings, in decreasing order of performance, are Random Forest, Naïve Bayes, KNN, and SVM. This hierarchy underscores the enhanced capabilities of the LCS-KELM classifier, which not only outperformed the traditional models but set a new benchmark in accuracy for COVID-19 detection using ML techniques.

Table 6. Comparative analysis of segmentation performance with existing techniques for 5-fold and 10-fold validation

Method	**5-Fold**			**10-Fold**		
	Accuracy (%)	**Sensitivity (%)**	**Specificity (%)**	**Accuracy (%)**	**Sensitivity (%)**	**Specificity (%)**
PCA+CNN [13]	70	70	63	75	76	68
DNN [23]	68	68	75	74	74	79
HQCNN [25]	78	73	68	80	77	72
Deep CNN [26]	80	70	77	84	76	81
Stacked ensemble model [27]	85	80	83	88	86	86
Proposed LCS-KELM classifier	92	90	90	94.7	94.9	94

Table 7. Comparative analysis of feature extraction performance with existing methods for 5-fold and 10-fold validation

Method	**5-Fold**			**10-Fold**		
	Accuracy (%)	**Sensitivity (%)**	**Specificity (%)**	**Accuracy (%)**	**Sensitivity (%)**	**Specificity (%)**
PCA+CNN [13]	72	76	63	78	87	68
DNN [23]	85	76	75	80	85	72
HQCNN [25]	81	88	78	83	86	73
Deep CNN [26]	85	78	79	81	82	80
Stacked ensemble model [27]	91	87	82	89	89	83
Proposed LCS-KELM classifier	96.4	92	90.4	96	96	94

Table 8. Comparative analysis of classification performance with existing techniques for 5-fold and 10-fold validation

Method	**5-Fold**			**10-Fold**		
	Accuracy (%)	**Sensitivity (%)**	**Specificity (%)**	**Accuracy (%)**	**Sensitivity (%)**	**Specificity (%)**
PCA+CNN [13]	75	78	67	77	84	72
DNN [23]	79	76	71	83	83	76
HQCNN [25]	81	84	80	86	85	79
Deep CNN [26]	85	80	76	87	86	82
Stacked ensemble model [27]	91	85	84	90	90	82
Proposed LCS-KELM classifier	96.4	93	92	96	95.7	94.1

Table 9. Comparative analysis with machine learning models

Method	**Fold**	**Decision Tree [20]**	**Random forest [19]**	**Naïve Bayes [22]**	**KNN [21]**	**SVM [18]**	**Proposed LCS-KELM**
Accuracy (%)	5	56	60	80	76	84	89.6
	10	67	64	63	70	89	96.3

Table 10. Comparative analysis with machine learning models

Method	**Fold**	**k-means clustering [41]**	**FCM [42]**	**MFCM [43]**	**Proposed TLMFCM**
Run Time (s)	5	9.8	9	8.3	6.4
	10	11	10.5	10	8

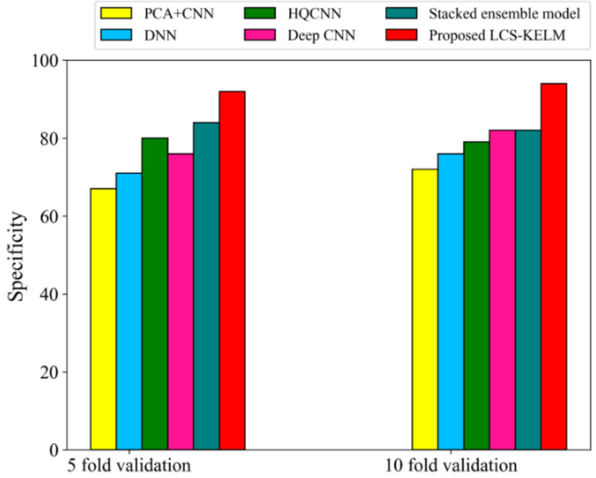


Figure 19. Classification specificity comparison

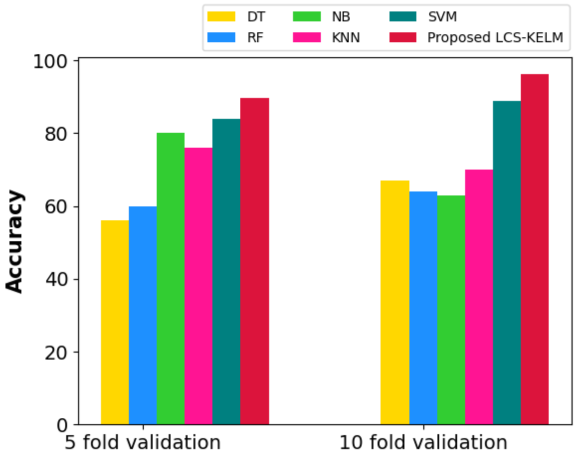


Figure 20. Accuracy analysis of classifier with baseline techniques

Table 9 presents a comparative analysis of the performance of different ML models, including Decision Tree [20], Random Forest [19], Naïve Bayes [22], KNN [21], SVM [18], and the proposed LCS-KELM, in the context of COVID-19 detection using CT scan data. The accuracy, expressed as a percentage, is used as the evaluation metric for these models in both 5-fold and 10-fold cross-validation scenarios. In the 5-fold validation, the Decision Tree model achieved 56% accuracy, Random Forest reached 60%, Naïve Bayes attained 80%, KNN scored 76%, and SVM showed 84%. Impressively, the proposed LCS-KELM outperformed all, achieving an accuracy of 89.6%. Similarly, in the 10-fold validation, the LCS-KELM model showcased its superiority with an accuracy of 96.3%, while the other models scored lower, highlighting the exceptional diagnostic precision offered by the proposed methodology for COVID-19 detection using CT scans.

Figure 21 provides a visual representation that offers a comparative analysis of the time consumed by various segmentation algorithms, including the traditional k-means clustering [41], FCM [42], and Modified FCM (MFCM) [43]. Notably, the TLMFCM stands out for its efficiency, taking only 5.9 seconds for a 5-fold cross-validation and 8.8 seconds for a 10-fold validation. This enhanced speed can be attributed to TLMFCM's distinctive ability to select optimal parameters during each iteration. Instead of adhering to fixed parameters or necessitating manual adjustments, TLMFCM dynamically fine-tunes its parameters, ensuring that it processes the dataset in a more streamlined manner. The consequence is a noticeable reduction in computational time without compromising on accuracy. Furthermore, such efficiency gains are crucial, especially in medical imaging where timely decisions can significantly impact patient outcomes. The detailed numerical results, which further substantiate these observations, are comprehensively laid out in Table 9. This table offers a quantitative breakdown, highlighting the edge TLMFCM has over traditional algorithms in terms of processing speed.

Table 10 provides a comparative analysis of the runtime performances of various ML models against the proposed TLMFCM when applied to CT scan data for COVID-19 detection. The table measures the runtime in seconds for each methodology under two different fold validations: 5-fold and 10-fold. For the 5-fold validation, the k-means clustering model [41] took 9.8 seconds, the FCM method [42] recorded a time of 9 seconds, the Modified Fuzzy C-Means (MFCM) [43] finished in 8.3 seconds, whereas the proposed TLMFCM showcased enhanced efficiency, completing the task in only 6.4 seconds. The 10-fold validation, which inherently demands more computation due to increased iterations, presented longer runtimes across the board. Here, the k-means clustering took 11 seconds, FCM was slightly quicker at 10.5 seconds, MFCM clocked in at 10 seconds, and again, the proposed TLMFCM outperformed the others, taking just 8 seconds. This data underscores the computational efficiency of the proposed TLMFCM method in comparison to its contemporaries.

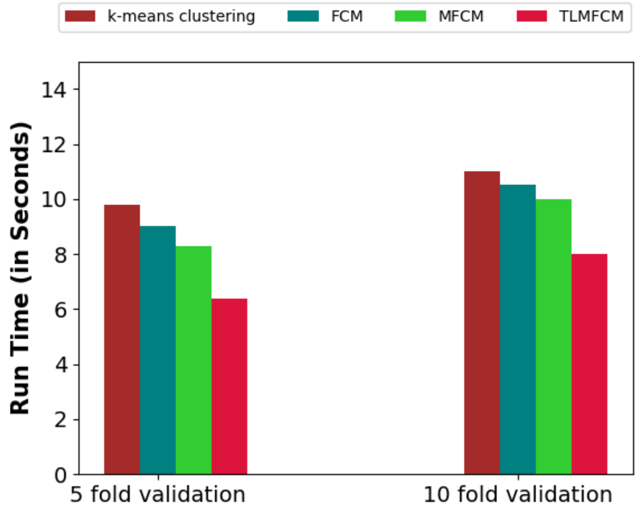


Figure 21. Time complexity analysis

4.7 Limitations of proposed approach

While our results are promising but certain limitations exists: The model's real-world performance across diverse global populations and different imaging equipment remains to be evaluated. The possibility of overfitting to the training data can't be completely ruled out without extensive testing on diverse datasets. As with all ML models, the quality and quantity of the training data play a pivotal role. Biases in data can inadvertently affect the model's predictions. Continuous updates and monitoring will be essential, especially considering the evolving nature of the COVID-19 virus and its manifestations in CT scans.

5. CONCLUSION AND FUTURE SCOPE

In conclusion, the core objective of our research was to devise an innovative and integrated methodology for the accurate detection of COVID-19 using CT scans. Through a synergetic combination of sophisticated segmentation techniques, notably TLMFCM, and the prowess of SSAE for robust feature extraction, followed by the precision of LCS-KELM for classification, we sought to address the multifaceted challenges that plague contemporary medical imaging diagnostics. Our results indicate that this methodology, when deployed in clinical settings, possesses the potential to drastically enhance the accuracy, speed, and efficiency of COVID-19 diagnosis, laying the foundation for prompt patient treatment and more effective containment measures. The broader significance of this research transcends mere technological advancement; it touches upon the very paradigms of clinical practice. By potentially reducing both false positives and negatives, we are looking at a future where clinical decisions are more informed, patient outcomes are improved, and the overall spread of the virus could be better managed. However, it is essential to remain grounded in recognizing the inherent limitations of this study. While we have made strides in a controlled environment, the actual litmus test lies in the diverse, global clinical settings, each with its unique challenges ranging from varied imaging equipment to differing patient demographics. As we look forward to the future, several areas demand attention and refinement. The optimization algorithms could be made even more robust, and integrating adaptive real-time learning could ensure the system remains relevant in the face of evolving disease patterns. Practical deployment, too, requires a nuanced approach. Beyond just technological integration, we must consider aspects like data privacy, the ever-present challenges of system interoperability, and the need for continuous model training. Additionally, the human factor cannot be overlooked; clinicians and other healthcare professionals will need to interface seamlessly with this technology, necessitating intuitive design and comprehensive training. To encapsulate, this research presents a promising pathway in the fight against COVID-19, harnessing the power of technology for healthcare betterment. As we progress, our objective remains clear: to make this methodology not just technologically advanced but also practically adoptable, ensuring a real-world impact that aligns with the broader goals of patient care and public health.

REFERENCES

[1] Lai, C.C., Shih, T.P., Ko, W.C., Tang, H.J., Hsueh, P.R. (2020). Severe acute respiratory syndrome coronavirus 2 (sars-cov-2) and coronavirus disease-2019 (covid-19): The epidemic and the challenges. International Journal of Antimicrobial Agents, 55(3): 105924. https://doi.org/10.1016/j.ijantimicag.2020.105924

[2] Nour, M., Cömert, Z., Polat, K. (2020). A novel medical diagnosis model for COVID-19 infection detection based on deep features and Bayesian optimization. Applied Soft Computing, 97: 106580. https://doi.org/10.1016/j.asoc.2020.106580

[3] Minaee, S., Kafieh, R., Sonka, M., Yazdani, S., Soufi, G.J. (2020). Deep-covid: Predicting covid-19 from chest x-ray images using deep transfer learning. Medical Image Analysis, 65: 101794. https://doi.org/10.1016/j.media.2020.101794

[4] Mishra, N.K., Singh, P., Joshi, S.D. (2021). Automated detection of COVID-19 from CT scan using convolutional neural network. Biocybernetics and Biomedical Engineering, 41(2): 572-588. https://doi.org/10.1016/j.bbe.2021.04.006

[5] Tabik, S., Gómez-Ríos, A., Martín-Rodríguez, J.L., Sevillano-García, I., Rey-Area, M., Charte, D., Herrera, F. (2020). COVIDGR dataset and COVID-SDNet methodology for predicting COVID-19 based on Chest X-Ray images. IEEE Journal of Biomedical and Health Informatics, 24(12): 3595-3605. https://doi.org/10.1109/JBHI.2020.3037127

[6] Nguyen, T.T., Nguyen, Q.V.H., Nguyen, D.T., Hsu, E.B., Yang, S., Eklund, P. (2020). Artificial intelligence in the battle against coronavirus (COVID-19): A survey and future research directions. arXiv preprint arXiv:2008.07343. https://doi.org/10.48550/arXiv.2008.07343

[7] Joshi, R.C., Yadav, S., Pathak, V.K., Malhotra, H.S., Khokhar, H.V., Parihar, A., Kohli, N., Himanshu, D., Garg, R.K., Bhatt, M.L., Kumar, R.A. (2021). Deep learning-based COVID-19 automatic diagnostic framework using chest X-ray images. Biocybernetics and Biomedical Engineering, 41(1): 239-254. https://doi.org/10.48550/arXiv.2008.07343

[8] Murugan, R., Goel, T., Mirjalili, S., Chakrabartty, D.K. (2021). WOANet: Whale optimized deep neural network for the classification of COVID-19 from radiography images. Biocybernetics and Biomedical Engineering, 41(4): 1702-1718. https://doi.org/10.1016/j.bbe.2021.10.004

[9] Punn, N.S., Agarwal, S. (2021). Automated diagnosis of COVID-19 with limited posteroanterior chest X-ray images using fine-tuned deep neural networks. Applied Intelligence, 51(5): 2689-2702. https://doi.org/10.1007/s10489-020-01900-3

[10] Shoeibi, A., Khodatars, M., Alizadehsani, R., Ghassemi, N., Jafari, M., Moridian, P., Srinivasan, D. (2020). Automated detection and forecasting of covid-19 using deep learning techniques: A review. arXiv preprint arXiv:2007.10785. https://doi.org/10.48550/arXiv.2007.10785

[11] Kassania, S.H., Kassanib, P.H., Wesolowskic, M.J., Schneidera, K.A., Detersa, R. (2021). Automatic detection of coronavirus disease (COVID-19) in X-ray and CT images: A machine learning based approach. Biocybernetics and Biomedical Engineering, 41(3): 867-879. https://doi.org/10.1016/j.bbe.2021.05.013

[12] Huang, S., Yang, J., Fong, S., Zhao, Q. (2021). Artificial intelligence in the diagnosis of COVID-19: Challenges and perspectives. International Journal of Biological Sciences, 17(6): 1581. https://doi.org/10.7150/ijbs.58855

[13] Rasheed, J., Hameed, A.A., Djeddi, C., Jamil, A., Al-Turjman, F. (2021). A machine learning-based framework for diagnosis of COVID-19 from chest X-ray images. Interdisciplinary Sciences: Computational Life Sciences, 13(1): 103-117. https://doi.org/10.1007/s12539-020-00403-6

[14] Toğaçar, M., Ergen, B., Cömert, Z. (2020). COVID-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X-ray images using fuzzy color and stacking approaches. Computers in Biology and Medicine, 121: 103805. https://doi.org/10.1016/j.compbiomed.2020.103805

[15] Tuncer, T., Dogan, S., Ozyurt, F. (2020). An automated residual exemplar local binary pattern and iterative ReliefF based COVID-19 detection method using chest X-ray image. Chemometrics and Intelligent Laboratory Systems, 203: 104054. https://doi.org/10.1016/j.chemolab.2020.104054

[16] Aggarwal, L., Goswami, P., Sachdeva, S. (2021). Multi-criterion intelligent decision support system for COVID-19. Applied Soft Computing, 101: 107056. https://doi.org/10.1016/j.asoc.2020.107056

[17] Jain, G., Mittal, D., Thakur, D., Mittal, M.K. (2020). A deep learning approach to detect Covid-19 coronavirus with X-ray images. Biocybernetics and Biomedical Engineering, 40(4): 1391-1405. https://doi.org/10.1016/j.bbe.2020.08.008

[18] Karatzoglou, A., Meyer, D., Hornik, K. (2006). Support vector machines in R. Journal of Statistical Software, 15(9): 1-28. https://doi.org/10.18637/jss.v015.i09

[19] Sundaravadivelu, B., Santhanakrishnan, K. (2023). Breast cancer detection using comprising fuzzy C-means and artificial bee colony optimization segmentation and grading with random forest classifier. Traitement du Signal, 40(6): 2751-2759. https://doi.org/10.18280/ts.400635

[20] Mulani, A.O., Birajadar, G., Ivković, N., Salah, B., Darlis, A.R. (2023). Deep learning based detection of dermatological diseases using convolutional neural networks and decision trees. Traitement du Signal, 40(6): 2819-2825. https://doi.org/10.18280/ts.400642

[21] Zouhal, L.M., Denoeux, T. (1998). An evidence-theoretic k-NN rule with parameter optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 28(2): 263-271. https://doi.org/10.1109/5326.669565

[22] Zhang, H. (2004). The optimality of naive bayes. American Association for Artificial Intelligence, 1(2): 3.

[23] Mukherjee, H., Ghosh, S., Dhar, A., Obaidullah, S.M., Santosh, K.C., Roy, K. (2021). Deep neural network to detect COVID-19: One architecture for both CT scans and Chest X-rays. Applied Intelligence, 51(5): 2777-2789. https://doi.org/10.1007/s10489-020-01943-6

[24] Panwar, H., Gupta, P.K., Siddiqui, M.K., Morales-Menendez, R., Bhardwaj, P., Singh, V. (2020). A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-Scan images. Chaos, Solitons & Fractals, 140: 110190. https://doi.org/10.1016/j.chaos.2020.110190

[25] Houssein, E.H., Abohashima, Z., Elhoseny, M., Mohamed, W.M. (2021). Hybrid quantum convolutional neural networks model for COVID-19 prediction using chest X-Ray images. arXiv preprint arXiv:2102.06535. https://doi.org/10.1093/jcde/qwac003

[26] Abbas, A., Abdelsamea, M.M., Gaber, M.M. (2021). Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network. Applied Intelligence, 51(2): 854-864. https://doi.org/10.1007/s10489-020-01829-7

[27] Das, A.K., Ghosh, S., Thunder, S., Dutta, R., Agarwal, S., Chakrabarti, A. (2021). Automatic COVID-19 detection from X-ray images using ensemble learning with convolutional neural network. Pattern Analysis and Applications, 24: 1111-1124. https://doi.org/10.1007/s10044-021-00970-4

[28] Gaur, L., Bhatia, U., Jhanjhi, N.Z., Muhammad, G., Masud, M. (2021). Medical image-based detection of COVID-19 using deep convolution neural networks. Multimedia Systems, 29: 1729-1738. https://doi.org/10.1007/s00530-021-00794-6

[29] Munusamy, H., Karthikeyan, J.M., Shriram, G., Revathi, S.T., Aravind Kumar, S. (2021). FractalCovNet architecture for COVID-19 Chest X-ray image Classification and CT-scan image Segmentation. Biocybernetics and Biomedical Engineering, 41(3): 1025-1038. https://doi.org/10.1016/j.bbe.2021.06.011

[30] Jangam, E., Barreto, A.A.D., Annavarapu, C.S.R. (2021). Automatic detection of COVID-19 from chest CT scan and chest X-Rays images using deep learning, transfer learning and stacking. Applied Intelligence, 52: 2243-2259. https://doi.org/10.1007/s10489-021-02393-4

[31] Ni, Q., Sun, Z.Y., Qi, L., Chen, W., Yang, Y., Wang, L., Zhang, X., Yang, L., Fang, Y., Xing, Z., Zhou, Z., Yu, Y., Lu, G.M., Zhang, L.J. (2020). A deep learning approach to characterize 2019 coronavirus disease (COVID-19) pneumonia in chest CT images. European Radiology, 30(12): 6517-6527. https://doi.org/10.1007/s00330-020-07044-9

[32] Badriyah, T., Sakinah, N., Syarif, I., Syarif, D.R. (2019). Segmentation stroke objects based on CT scan image using thresholding method. In 2019 First International Conference on Smart Technology & Urban Development (STUD), Chiang Mai, Thailand, pp. 1-6. https://doi.org/10.1109/STUD49732.2019.9018825

[33] Chuang, K.S., Tzeng, H.L., Chen, S., Wu, J., Chen, T.J. (2006). Fuzzy c-means clustering with spatial information for image segmentation. Computerized Medical Imaging and Graphics, 30(1): 9-15. https://doi.org/10.1016/j.compmedimag.2005.10.001

[34] Zhao, C., Xu, Y., He, Z., Tang, J., Zhang, Y., Han, J., Shi, Y., Zhou, W. (2021). Lung segmentation and automatic detection of COVID-19 using radiomic features from chest CT images. Pattern Recognition, 119: 108071. https://doi.org/10.1016/j.patcog.2021.108071

[35] Meng, Q., Catchpoole, D., Skillicom, D., Kennedy, P.J. (2017). Relational autoencoder for feature extraction. In 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, pp. 364-371. https://doi.org/10.1109/IJCNN.2017.7965877

[36] Rashid, N., Hossain, M.A., Ali, M., Sukanya, M.I., Mahmud, T., Fattah, S.A. (2021). AutoCovNet: Unsupervised feature learning using autoencoder and feature merging for detection of COVID-19 from chest X-ray images. Biocybernetics and Biomedical Engineering, 41(4): 1685-1701. https://doi.org/10.1016/j.bbe.2021.09.004

[37] Ding, S., Xu, X., Nie, R. (2014). Extreme learning machine and its applications. Neural Computing and Applications, 25(3): 549-556. https://doi.org/10.1007/s00521-013-1522-8

[38] Dhiman, G., Kumar, V. (2019). Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. Knowledge-Based Systems, 165: 169-196. https://doi.org/10.1016/j.knosys.2018.11.024

[39] Li, X., Niu, P., Liu, J. (2018). Combustion optimization of a boiler based on the chaos and Levy flight vortex search algorithm. Applied Mathematical Modelling, 58: 3-18. https://doi.org/10.1016/j.apm.2018.01.043

[40] Dataset: https://www.kaggle.com/anaselmasry/covid19normalpneumonia-ct-images?select=COVID2_CT.

[41] Wu, M.N., Lin, C.C., Chang, C.C. (2007). Brain tumor detection using color-based k-means clustering segmentation. In Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007), pp. 245-250. https://doi.org/10.1109/IIHMSP.2007.4457697

[42] Li, Y., Shen, Y. (2010). Fuzzy c-means clustering based on spatial neighborhood information for image segmentation. Journal of Systems Engineering and Electronics, 21(2): 323-328. https://doi.org/10.3969/j.issn.1004-4132.2010.02.024

[43] Sikka, K., Sinha, N., Singh, P.K., Mishra, A.K. (2009). A fully automated algorithm under a modified FCM framework for improved brain MR image segmentation. Magnetic Resonance Imaging, 27(7): 994-1004. https://doi.org/10.1016/j.mri.2009.01.024