



## Identifying Severity of COVID-19 Medical Images by Categorizing Using HSDC Model

K. Ravishankar\* and C. Jothikumar

Department of Computing Technologies, School of Computing, SRM Institute of Science and Technology, Chengalpattu, Tamilnadu, 603203, India

\*Corresponding Author: K. Ravishankar. Email: rk1900@srmist.edu.in

Received: 08 December 2022; Accepted: 20 March 2023; Published: 26 May 2023

**Abstract:** Since COVID-19 infections are increasing all over the world, there is a need for developing solutions for its early and accurate diagnosis is a must. Detection methods for COVID-19 include screening methods like Chest X-rays and Computed Tomography (CT) scans. More work must be done on preprocessing the datasets, such as eliminating the diaphragm portions, enhancing the image intensity, and minimizing noise. In addition to the detection of COVID-19, the severity of the infection needs to be estimated. The HSDC model is proposed to solve these problems, which will detect and classify the severity of COVID-19 from X-ray and CT-scan images. For CT-scan images, the histogram threshold of the input image is adaptively determined using the ICH Swarm Optimization Segmentation (ICHSeg) algorithm. Based on the Statistical and Shape-based feature vectors (FVs), the extracted regions are classified using a Hybrid model for CT images (HSDC-CT) algorithm. When the infections are detected, it's classified as Normal, Moderate, and Severe. A fused FHI is formed for X-ray images by extracting the features of Histogram-oriented gradient (HOG) and Image profile (IP). The FHI features of X-ray images are classified using Hybrid Support Vector Machine (SVM) and Deep Convolutional Neural Network (DCNN) HSDC-X algorithm into COVID-19 or else Pneumonia, or Normal. Experimental results have shown that the accuracy of the HSDC model attains the highest of 94.6 for CT-scan images and 95.6 for X-ray images when compared to SVM and DCNN. This study thus significantly helps medical professionals and doctors diagnose COVID-19 infections quickly, which is the most needed in current years.

**Keywords:** CT-scan; convolution neural network (CNN); deep CNN (HSDC); hybrid support vector machine (SVM); improved chicken swarm optimization (ICHO); COVID-19 and image profile (IP)



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1 Introduction

COVID-19 has infected many people, leading to the cause of dangerous pneumonia. Fever, tiredness, high cough, and breathlessness are its symptoms. Severe infections of this disease may cause renal failure leading to death. Since this virus has unique properties, it won't be easy to detect it earlier. Hence it has to be detected well in advance and adequately controlled for the speedy recovery of patients [1].

The detection methods of COVID-19 include screening techniques such as CT-scan and chest X-rays which provide visual guidance. These techniques are faster and more efficient when compared to other tests [2]. But frequently taking CT scans is not healthy for kids and pregnant women because high radiation may affect them severely [3]. Recently, Deep Learning (DL) techniques have been utilized in medical applications effectively for detecting various diseases so that the detected diseases may reduce the burden for medical practitioners [4].

Implementing the DL algorithm in various medical applications provides very effective prediction and diagnosis and helps control the spread of COVID-19 [5,6]. Early stages of detecting and discovering viruses with high accuracy can reduce the spreading of the virus, and estimating the severity may also reduce the death ratio. Except for a few works, many still need to apply DL techniques in X-ray and CT-scan images. Before training, usually, these images are in grayscale. Hence the preprocessing method is essential for evaluating the images since their size and structure may vary.

Localizing the infection and segmenting the pneumonia lesions is challenging for accurate detection and further actions. But most works rarely identify the shape and size of the infected regions from pneumonia lesions. Segmentation of CT scan medical images would be beneficial to prioritize the patients by severity. It will help to localize the infected region and provide its spatial features like size and shape. But only a few works have applied segmentation tasks for COVID-19 CT images in the chest.

The problem statement can be defined as follows: "There are little works on preprocessing the datasets such as eliminating the diaphragm portions, enhancing the image contrast, and minimizing the noises. In addition to detecting COVID-19 infections, the severity of the infection needs to be estimated using the DL techniques. The infected regions have to be identified using segmentation techniques".

This paper designs an HSDC model for detecting and classifying COVID-19 from Chest X-rays and severity from CT images. The major contributions of HSDC models are:

- (i) The HSDC model estimates the severity of infection from CT images and classifies the infections from X-ray images by applying hybrid DL methods.
- (ii) It applies an optimization technique for adaptively determining the segmentation threshold for CT images.

The content of this manuscript is structured as follows. Section 2 presents a brief literature review of existing works, Section 3 includes a brief discussion of existing techniques used in the proposed work, Section 4 contains the proposed methodology of the HSDC model, and Section 5 presents the simulation results. Section 6 concludes the paper along with future work.

## 2 Related Works

In [7], VGG19 and VGG16 models are used to classify the X-ray images as present or absent for COVID-19. During the classification step, these two models were trained using the transfer learning

method with pre-trained weights of ImageNet. But, this work did not determine the severity of the infection.

In the COVID-19 detection model of Ozturk et al. [8], both binary classifiers (COVID-19 and Normal) and multi-class classifiers (COVID-19, Normal, and Pneumonia) were used. This model achieved high classification accuracy for both these classes. However, this model did not use any feature extraction procedure. Alotaibi et al. [9] have developed a DL method for detecting COVID-19 infection from chest X-ray images—three types of classes based on the classification experiments. Moreover, the Transfer-learning method is applied to boost the model's performance.

Aminu et al. [10] proposed a DL architecture CovidNet for deep feature extraction, which uses grayscale images with a limited training dataset. Unlike the other approaches, classification is done for both X-ray and CT-scan images.

A COVID-19 identification technique was developed from CT images and X-rays [11]. After performing the preprocessing techniques like data cleaning and image augmentation, the CNN model was utilized for classifying the disease. They have compared their performance against the Inception V3, Xception, and ResNeXt models. However, the prediction accuracy could be better.

A U-Net architecture has been designed by Qiblaweya et al. [12], which segments the infected CT images and marks the infected regions. But noise reduction was not performed in this method. A hierarchical model was designed in [12] for COVID-19 detection. The DL-based Encoder-Decoder was applied in the CNN model to classify the infection as severe, normal, and mild. But noise reduction was not performed in this method.

Müller et al. [13] have developed an automated pipeline for segmenting the COVID-19-infected regions. This technique generates unique and random image slices for training and utilizes data augmentation procedures for training. They have used a 3D U-Net DL model to solve the overfitting issues.

Arowolo et al. [14] have applied an ML method for COVID-19 detection Using KPCA-SVM. It uses the latest cases and their gene data to extract suitable features. In [15], they used ABC and SVM classifiers for COVID-19 detection in an IoT data system on a San Francisco COVID-19 dataset. Again in [16], they used ANN and CNN classifiers for detection from chest X-ray images.

Table 1 presents the comparison of existing approaches with their advantages and drawbacks.

**Table 1:** Comparison of existing approaches

Author	Proposed method	Advantages	Drawbacks
Daniel Arias-Garzón et al. (2021)	VGG19 and VGG16 models are used for classification using a transfer learning scheme with pre-trained weights from ImageNet.	This model achieved high classification.	This work did not determine the severity of the infection.

(Continued)

**Table 1: Continued**

Author	Proposed method	Advantages	Drawbacks
Tulin Ozturk et al. (2021)	The proposed method used binary and multi-class classifiers (Covid and Normal).	This model achieved high classification accuracy for both these classes.	This model did not use any feature extraction procedure.
Munif Alotaibi et al. (2021)	The authors developed a DL method for COVID-19 detection from X-ray images.	This model has achieved higher accuracy for classification.	Diaphragm regions were not separated, and the image noise has not been minimized.
Muhammad Aminu et al. (2021)	CNN model was utilized for classifying the disease. They have compared their performance against the various models.	This model achieved high classification.	Diaphragm regions were not separated, and the image noise has not been minimized.
Rachna Jain et al. (2021)	V3, Xception, and ResNeXt	Data cleaning and augmentation steps were performed during preprocessing.	This method does not claim any medical accuracy.
Yazan Qiblaweya et al. (2021)	The authors have segmented the CT scan images and marked the infected regions.	This model achieved high classification accuracy	But noise reduction was not performed in this method
Dominik Müller et al. (2021)	The authors have developed an automated pipeline for segmentation.	They have used a 3D U-Net DL model to solve the overfitting issues.	Image noise is not reduced.
Micheal Olaolu Arowolo et al. (2022)	The authors have applied the KPCA-SVM method for the prediction of COVID-19.	It uses the latest cases and their gene data to extract relevant features.	But noise reduction was not performed in this method
Micheal Olaolu Arowolo et al. (2022)	The authors have used ABC and SVM to detect COVID-19 in IoT-based systems.	It uses hybrid ML, and DL approaches for higher accuracy	However, this work did not determine the severity of the infection.
Micheal Olaolu Arowolo et al. (2022)	The authors have used ANN and CNN techniques for detection.	It uses hybrid ML, and DL approaches for higher accuracy	However, this work did not determine the severity of the infection.

### 3 Existing Techniques Used in this Model

#### 3.1 OTSU Thresholding for Segmentation

In Image processing, selecting a suitable threshold value of gray-level pixels for extracting the features from their background is essential. Plenty of techniques have been developed for this purpose. OTSU is a threshold-based image segmentation technique. It uses a threshold selection rule based on the max Inter-class difference of the background from the given image. Another way to find the optimal thresholding is through the Ostu threshold. They use a variance-based method, finding the threshold value where the weighted variation between the background and foreground pixels is the least.

In OTSU [17], an optimal threshold is selected using the splitting condition to separate the resultant classes in gray levels. After implementing the threshold techniques in an image, pixel by grey level histogram representing the probability distribution is applied in several steps to return the final output as total inter-class variance  $G(T)$ . It is not only a standard technique in image processing but also applied in unsupervised learning solutions of pattern recognition. Let the pixels of a given image with  $L$  gray levels can be given by

Let the pixels of a given image with  $L$  gray levels can be given by

$$N = p_0 + p_1 + \dots + p_L \quad (1)$$

Then grey level histogram is represented as a probability distribution as

$$PR(i) = p_i/N, i = 1, 2 \dots L \quad (2)$$

A threshold  $T$  divides the pixels of the grey scale image into the background and the object classes  $C_1 = (0, 1, 2 \dots T)$  and  $C_2 = (T + 1, T + 2, T + 3, \dots L)$ , respectively.

The probabilities  $C_1$  and  $C_2$  are given by

$$Pr_1 = Pr(T) = p_1 + p_2 + p_3 + \dots + p_T \quad (3)$$

$$Pr_2 = 1 - Pr(T) = p_{T+1} + p_{T+2} + p_{T+3} + \dots + p_L \quad (4)$$

The mean values of  $C_1$  and  $C_2$  are given by

$$M_1 = (1.p_1 + 2.p_2 + 3.p_3 + \dots + T.p_T)/Pr_1 \quad (5)$$

$$M_2 = ((T + 1).p_{T+1} + (T + 2).p_{T+2} + \dots + L.p_L)/Pr_2 \quad (6)$$

Then the value of  $T$  can be determined from the following parameters

$$A(T) = [Pr_1, Pr_2, (M_2 - M_1)^2] \quad (7)$$

$$B(T) = Pr_1 \left( \sum_{i=1}^T (i - M_1)^2 .p_i/Pr_1 \right) + Pr_2 \left( \sum_{i=T}^L (i - M_2)^2 .p_i/Pr_2 \right) \quad (8)$$

$$G(T) = A(T) + B(T) \quad (9)$$

Here  $A(T)$  represents the Inter-class variance where in terms of statistical characteristics first order,  $B(T)$  the other one is termed as inter-class variance statistical characteristics of second order, and  $G(T)$  is the total inter-class variance.

### 3.2 Improved Chicken Swarm Optimization (ICHO) Algorithm

ICHO [18] is a NIB algorithm as New Intelligent Bionic, behavioral of a chicken swarm searching for food in nature. Compared to the traditional CHO algorithm, the improved version included the Chick position update mode and the cock and hen position update modes.

It is demonstrated that ICHO is faster and more cost-effective than other optimization methods. Unlike traditional optimization methods, it does not require the problem to be differentiable. It includes updating Cock position, Hen position, chick, and population.

#### (i) Update of Cock Position

When the cock is away from the mid-point  $p$  of the hen in a cluster, it will perform a random search in a more extensive region, thus improving the probability of global tracking. On the other hand, when the cock is nearer to the mid-point  $p$  of the hen, it will perform a dynamic search in a minor region, which improves the local estimation capacity of ICHO and the chances of determining the best solution [19].

The cock position update equation is given as

$$P_i^C(t+1) = QP_i^C(t) (1 + Z(-1, \alpha^2)) \quad (10)$$

where  $C$  is an individual cock,  $Q = \gamma e^d$ , where  $d$  is the distance between the  $i^{\text{th}}$  group and position of cock, is a proportional constant.  $Z(-1, \alpha^2)$  is Gaussian spreading having  $-1$  as mean and  $\alpha^2$  as variance.

$$\alpha^2 = \exp\left(\frac{f_i - f_k}{|f_i| + \partial}\right) i, k \in [1, 2, \dots, N], k \neq i \quad (11)$$

$k \in [1, 2, \dots, N]$  and  $i$  denote cock's indices, dynamically chosen from the cock groups,  $i \neq k$ ,  $f_i$ , and  $f_k$  represent the fitness function of the  $i^{\text{th}}$  &  $k^{\text{th}}$  cocks, respectively.  $\partial$  A small constant evades the divide to zero error;  $\gamma$  is the proportional coefficient.

#### (ii) Hen Position Update Mode

Among the initial population,  $G_{\text{best}}$  individuals with better fitness values are selected, and one is randomly chosen as the supreme individual. During the foraging step, the supreme individual is considered the learning objective for the hen.

Hence, using the following equation, the hen (H) position is updated based on the supreme individual.

$$P_i^H(t+1) = P_i^H(t) * S_1 * \text{rand} * (P_{r_1}^C(t) - P_i^H(t)) + S_2 * \text{rand} * (P_{r_2}(t) - P_i^H(t) + \text{rand} * (P_e(t) - P_i^H(t))), e \in [1, 2, \dots, \text{hao}] \quad (12)$$

where  $C$  and  $H$  denote the cock and hen,  $r_1$  is the index of cock where  $i^{\text{th}}$  hen's group mate,  $r_2$  is the index of the Chicken,  $P_e(t)$  is the chosen member hen's supreme learning object in the  $t^{\text{th}}$  round,  $\text{hao}$  denotes the Elite candidates fetched from the population.

$S_1$  and  $S_2$  are calculated as

$$S_1 = e^{(f_i - f_{r_1}) / \text{abs}(f_i)} + \delta \quad (13)$$

$$S_2 = e^{(f_{r_2} - f_i)} \quad (14)$$

where  $f_i$ ,  $f_{r_1}$ , and  $f_{r_2}$  represent the fitness function values at  $i$ ,  $r_1$ , and  $r_2$ , respectively.

### (iii) Chick Position Update Mode

The chick's position can be updated based on the position of the chick's mother and cock, which is  $i^{\text{th}}$  chick's group mate.

### (iv) Population Update Strategy

In most of the optimization solutions, during higher iterations, the number of similar individuals may linearly increase, thereby reducing the chances of meeting the optimal global solution. Hence, to keep the population's diversity unaffected, the population's individuals are updated.

### 3.3 MADF Technique

This technique [19] eliminates the noises from the X-ray dataset. It removes the noises from the images after performing several iterations where the edge portions are removed. In this technique, the input images are divided into slices. These slices are filtered one by one and then fused again.

An Input image 'O' mixed with multiplicative noise, also called a speckle, n is denoted as

$$O_0 = O.n \quad (15)$$

$$n = \frac{O - I}{\sqrt{I}}, \text{ and } I \text{ is the noise intensity}$$

The binding between the image and its noise should be minimized. This noise elimination process is iteratively executed as long as the noisy segment of the image does not reach the Gaussian term.

The kurtosis (ku) is computed by

$$ku = \frac{\frac{1}{N} \sum_{i=0}^N (I - u)^4}{\left[ \frac{1}{N} \sum_{i=0}^N (I - u)^2 \right]^2} - 3 \quad (16)$$

To obtain an accurate image, the filtering procedure is iteratively performed until the value k is equal to 0.001 since all the features will be eliminated if k becomes 0.

### 3.4 SVM for Classification

The fundamental aim of SVM is to determine the best hyperplane which maximally divides the target class into two. Geometrically, the SVM algorithm determines an optimal hyperplane with the maximal margin to divide the two classes,

The training set of SVM is represented as

$$(p_1, q_1), (p_2, q_2), \dots, (p_n, q_n), \quad p_j \in R^n, \quad q_j \in \{+1, -1\}$$

Here  $p_j$  is the characteristic vector of the model as input, and  $q_j$  is the output catalog = +1 or -1. SVM divides the positive and negative instances using a hyperplane as

$$w.p + b = 0, \quad w \in R^n, \quad b \in R \quad (17)$$

Here  $w.p$  indicates the dot product of  $w$  and  $p$ .

The Radial Basis Function (RBF) is a Kernel part used to find a non-linear classifier or regression line. The RBF kernel function for  $(P_1, P_2)$ , which calculates the likeliness of these points, can be denoted

as:

$$\text{RBF}(P_1, P_2) = \exp\left(-\frac{\|P_1 - P_2\|^2}{2\sigma^2}\right) \quad (18)$$

where  $\|P_1 - P_2\|$  is the distance between  $P_1$  and  $P_2$  and  $\sigma^2$  is the variance.

Tables 2 and 3 provide the abbreviations and notations used in the proposed methodology.

---

### ***SVM Training Algorithm***

---

**Input:** D(X, Y): X-input features, Y-classified labels,

LR-learning rate

N-Number of iterations

**Output:** Classified labels

---

1. Initialize LR randomly.
  2. **for** (LR = 1 to N)
  3. **Do**
  4.     Error = 0
  5. **for** (i in each X)
  6. **Do**
  7.     B = Y[i] \* X[i]
  8.     **if** ((B \* w) < 1)**then**
  9.         Update w = w + LR \* B \* (-2 \* (1/N) \* w)
  10.     **else**
  11.         Update w = w + LR \* (-2 \* (1/N) \* w)
  12.     **End if**
  13. **End Loop**
  14. **End Loop**
- 

**Table 2:** Table of abbreviations

Abbreviation	Expansion
IP	Image Profile
HOG	Histogram-oriented Gradient
CNN	Convolution Neural Network
ICHO	Improved Chicken Swarm Optimization
CL	Convolutional Layer
FCL	Fully Connected Layer
RF	Random Forest
SVM	Support Vector Machine
DL	Deep Learning
ML	Machine Learning



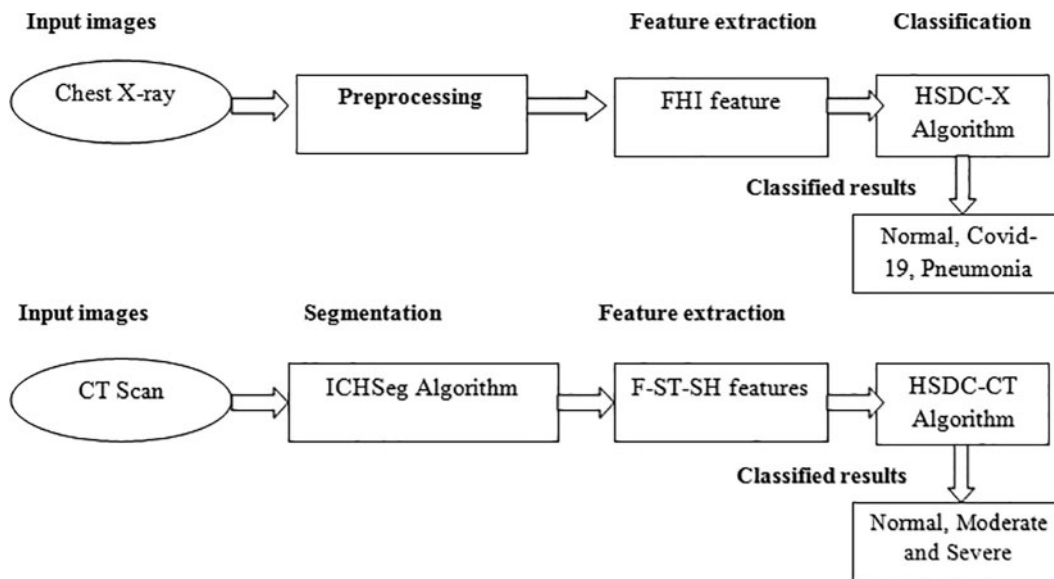
**Table 3:** Table of symbols and notations

Symbol	Explanation
$\alpha^2$	Variance
$p_j$	Characteristic vector
$w$	Weight constant
$C$	Mean value
$\gamma$	Proportional constant
$R$	Fitness function

## 4 Proposed Methodology

### 4.1 Overview

In this paper, the HSDC classification model is developed for the identification and severity estimation of COVID-19 from CT-scan and X-ray images. The HSDC model diagram is shown in Fig. 1. It contains HSDC-CT and HSDC-X algorithms for classification. The lung region of the CT-scan image is segmented using the ICHSeg algorithm. Fused statistical and shape (F-ST-SH) FVs are extracted from the segmented lung images. The F-ST-SH FV is then classified using the HSDC-CT algorithm as severe, moderate, and severe. The X-ray images are preprocessed using the existing MADF technique. After the preprocessing step, the FHI FVs are used for training the model. The HSDC-X algorithm classifies images of the X-ray as whether it's COVID-19 or else pneumonia and its normal condition.



**Figure 1:** Proposed model

## 4.2 COVID-19 Classification of CT-Scan Images

### 4.2.1 ICHO-Based Segmentation (ICHSeg)

Preprocessing is the preliminary phase in image processing which consists of Histogram and morphological techniques. The Lung CT scan image has four regions (i) Black background, (ii) a dark grayed circular area, (iii) a bright area (iv) darker gray-shaded lungs. Since the lung portion has to be segmented, the first two regions should be removed.

In this technique, the image's Histogram is constructed and analyzed to pick up a threshold, which detects and removes the image's outer portion. Let  $Im$  denotes the preprocessed CT image of size  $M \times N$ . From  $Im$ , the histogram  $H$  is estimated using the step value  $k$ , from which the threshold  $\eta$  is estimated.

Then the resultant image ( $ImR$ ) after removing the background is given by

$$ImR = Im(x, y) - T \quad (19)$$

The Otsu threshold  $T$  for each slice is computed Using the ICHSeg algorithm.

The Fitness function of ICHSeg is derived as

$$f_i = w_1.A(T) + w_2.B(T) + w_3.G(T) \quad (20)$$

where  $A(T)$ ,  $B(T)$  and  $G(T)$  are the threshold selection parameters defined by Eq. (1) and  $w_1$ ,  $w_2$  and  $w_3$  are weighting constants such that  $w_1 + w_2 + w_3 = 0$ .

Then segmentation is performed using the ICHSeg algorithm, which is illustrated below:

---

#### **ICHSeg Algorithm**

---

I-Input image, N-No of chickens,  $f_i'$ -new fitness,  $B_p$ -Best Chick Position

---

1. For each Input image  $I_k$
  2. Convert to a greyscale image
  3. Randomly initialize the population of N chickens
  4. Divide  $I_k$  into  $C_1$  and  $C_2$
  5. While (iter < MaxIter)
  6.     Estimate the fitness  $f_i$  of chickens Using Eq. (11)
  7.     Divide the swarm into various groups
  8.     For each,  $i$  in (1, 2 . . . N)
  9.         Update the cock position Using Eq. (2)
  10.        Update the hen position Using Eq. (4)
  11.        Update the chick position
  12.        Determine new fitness  $f_i'$
  13.        If ( $f_i'$  is better than  $f_i$ )
  14.            Update the Solution
  15.     End if
  16.     End For
  17.     End While
  18.  $T =$  Best chick position  $B_p$
  19. Find the resultant image  $ImR$  Using Eq. (10)
  20. Return the segmented lung regions
  21. End For
-

### 4.2.2 SHST Feature Extraction

Feature extraction is a key area for the classification of image properties. Moreover, it is utilized to enhance detection accuracy by choosing significant features. In [20], in the  $F_{SH}F_{ST}$  feature extraction phase, various shape and statistical features are extracted and fused from the segmented lung regions.

Lets  $\{x_1, x_2, x_3, \dots, x_n\}$  be the centers of the shapes which are fed as input to the region-growing method. It finds the associated n aspirant portions  $\{B_1, B_2, B_3, \dots, B_n\}$ .

The statistical features are represented by

$$F_{ST}(V_j) = \{F1_{av}, F2_{Me}, F3_{Mo}, F4_{Va}, F5_{SD}\}$$

where  $F1_{av}$ ,  $F2_{Me}$ ,  $F3_{Mo}$ ,  $F4_{Va}$ , and  $F5_{SD}$  represent the statistical properties Average, Median, Mode, Variance, and Standard deviation, respectively. These properties of aspirant regions are extracted from each nodule element portion  $B_i$  in a slice  $S_j$  and fused to get an  $FV_{vi}$ .

The shape feature is represented by

$$F_{SH}(V_j) = \{F6_{SCF}\}$$

$F6_{SCF}$  has estimated from the lesion's shape, and appearance in the image's adjacent slices.

Then the  $F_{SH}F_{ST}$  vector for the slice  $S_j$  is given by

$$F_{SH}F_{ST}(V_j) = \sum_i [F_{SH}(V_i), F_{ST}(V_i)] \tag{21}$$

### 4.2.3 Classification Using HSDC-CT

The extracted regions are classified using the HSDC-CT algorithm depending on the FV. The HSDC-CT classifier combines the advantages of SDC [SVM, DCNN] classifiers [21,22]. The database containing the extracted features is split into training and testing sets.

#### DCNN Architecture

DCNN applies filters on the image's actual pixels to collect patterns' details using a traditional neural network. The DCNN model proposed in this study is presented in Fig. 2. The layers of DCNN are described below:

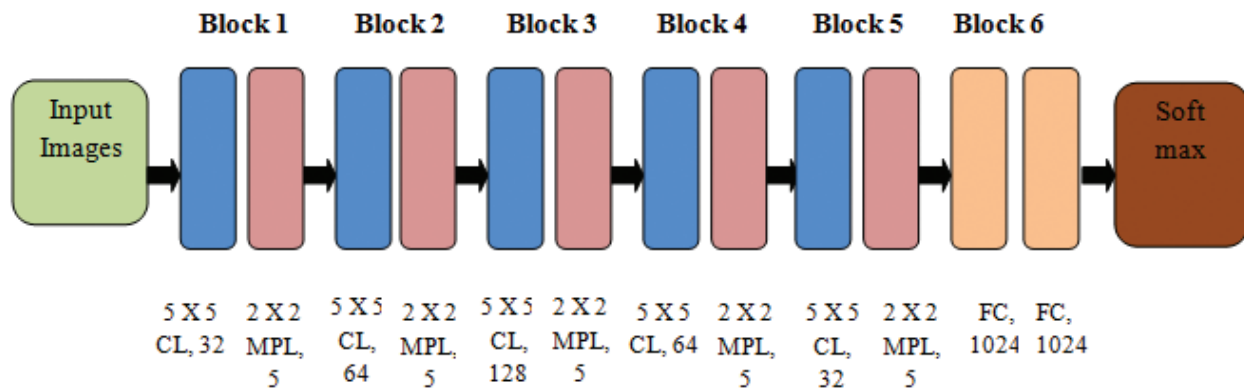


Figure 2: The DCNN model

**Convolutional layer (CL):** In this layer,  $n$  filters are applied to each detected feature map in the input. A Rectified Linear Unit (ReLU) activation function changes the network into a non-linear one.

**Max Pooling layer (MPL):** The conventional technique is to split and detect the feature mapped as sub-slices of size  $2 \times 2$ . It mainly retains the optimum values.

**Fully Connected Layer (FCL):** In FCL, the entire neurons of the previous layers are connected to the next ones. The labels are categorized depending on the outputs from CLs and reduced with MPL.

Table 4 describes the DCNN model depicted in Fig. 2.

**Table 4:** DCNN parameters

Layer	Count	Weight/Filter size	Activation function
Input			
CL	1	$32:5 \times 5$	ReLU
MPL	1	$5:2 \times 2$	ReLU
CL	2	$64:5 \times 5$	ReLU
MPL	2	$5:2 \times 2$	ReLU
CL	3	$128:5 \times 5$	ReLU
MPL	3	$5:2 \times 2$	ReLU
CL	4	$64:5 \times 5$	ReLU
MPL	4	$5:2 \times 2$	ReLU
CL	5	$32:5 \times 5$	ReLU
MPL	5	$5:2 \times 2$	ReLU
FCL	1	1024	ReLU
FCL	1	1024	SoftMax

#### 4.2.4 HSDC-CT

The extracted FV is passed to the HSDC-CT algorithm, which detects the COVID-19-infected images and classifies them into normal, moderate, and severe. CNN is the primary DL model which can produce predictive FVs from labeled inputs. SVM is an unsupervised algorithm that learns a decision function for detection.

The steps involved in the Hybrid model are summarized in the following algorithm:

<b>HSDC-CT Algorithm</b>	
Input: Train_vector $\{Tr F_{ST} F_{SH} V\}$ , Test_vector $\{Te F_{ST} F_{SH} V\}$ and Labels	
Output: Classification results	
1.	For all $trv \in \{Te_{FHH} V\}$
2.	Size_F = Sizeof (Tr $F_{ST} F_{SH} V$ )
3.	Fetch the class values
4.	S = Sizeof(Te $F_{ST} F_{SH} V$ )
5.	End for
6.	For each $trv(i, j)$ , $i = 1$ to $m$ , $j = 1$ to $n$
7.	Train <sub>i</sub> = $trv(i, j)$

(Continued)

---

Continued

---

8. uniclass = distinct(Label)
  9. L = Sizeof(uniclass)
  10. End For
  11. If (Label is a unique class of iTrain)
  12. Class\_new = Label
  13. End if
  14. SVM struct = Svmtrain(Tr  $F_{ST} F_{SH} V$ , Class\_new, 'svm kernel', 'rbf')
  15. Determine Tr  $F_{ST} F_{SH} V$  Using the SVM struct
  16. Obtain Tr1(X) and Trst1(X) from Tr  $F_{ST} F_{SH} V$  and Train<sub>t</sub>
  17. Train\_CNN(CL, Sub-sampling layer)
  18. Bz = batch size(trv)
  19. CNN\_result = test\_y(cnn\_output)
  20. Return → classification results (Severe, Moderate, and Normal)
  21. Stop
- 

Each feature of the image is picked up from the test set of CT image features. The training determines the dimension of the feature matrix. Then SVM Struct Uses the SVM kernel and RBF functions. Then the SVM training algorithm estimates the training features. In the next phase, CNN is trained, and the batch size of the testing feature is computed. The neurons from the CNN output layer are returned as the classification results as severe, moderate, and very severe.

---

**Pseudo code of HSDC-CT**


---

1. Fetch the testing CT images Te
  2. Get the FVs  $F_{ST} F_{SH}(V_j)$  of Te Using [Eq. \(12\)](#)
  3. Compute the size of the training FV matrix tr
  4. Read the class values of tr
  5. Compute the size of the testing vector matrix tev
  6. Get the unique class with length N of each tev
  7. Obtain the class Train\_T of each training feature vector trv
  8. If Label is a unique class of the training vector,
  9. Train each trvUsingsvm and rbf kernel functions
  10. Obtain the training class of each trv and tevUsing the SVM training algorithm
  11. Train CNN Using CL and Sub-Sampling
  12. Calculate the batch size of tev
  13. Find out neurons from the output layer of CNN
  14. Return the classification results as {Severe, Moderate, and Normal}
- 

### 4.3 COVID-19 Detection from X-ray Images

This process has the FHI feature extraction and HSDC-X classification phases.

#### 4.3.1 FHI Feature Extraction

When the preprocessing procedure is completed, the HOG and IP features of images are extracted. Then they are fused into an FHI feature, resulting in a rich set of features for observation. The FHI features are fed as input to the training process during the classification process.

The steps involved in HOG feature extraction are described below:

- Step-1: Get the Input X-ray Image  
 Step-2: Preprocess the image Using MADF as illustrated in Section 3.2  
 Step-3: Generate the Gradient Image  
 Step-4: Divide the Image window into cells and overlapping blocks  
 Step-5: Compute the HOG for each block and perform normalization  
 Step-6: Concatenate all normalized histograms  
 Step-7: Obtain the HOG feature vector

The features are obtained by extracting the IPs. The accuracy of rebuilt images is measured by inspecting the IPs of sharp edges.

The feature selection and fusion process was illustrated in the following equations [21]

$$f_{HOG_{1 \times n}} = \{HOG_{1 \times 1}, HOG_{1 \times 2}, HOG_{1 \times 3}, \dots, HOG_{1 \times n}\} \quad (22)$$

$$f_{IP_{1 \times m}} = \{IP_{1 \times 1}, IP_{1 \times 2}, IP_{1 \times 3}, \dots, IP_{1 \times m}\} \quad (23)$$

Then  $f_{HOG}$  and  $f_{IP}$  are fused Using the concatenation operation given by

$$FHI(V_j) = \{HOG_{1 \times 1} \oplus IP_{1 \times 1}, HOG_{1 \times 2} \oplus IP_{1 \times 2}, \dots, HOG_{1 \times n} \oplus IP_{1 \times m}\} \quad (24)$$

Eq. (24) denotes the FHI features.

Then HSDC algorithm classifies the X-ray chest images into different categories.

Here  $n = 3780$  and  $m = 300$ . Hence 4080 fused FHI features are obtained in which 20% are selected, depending on the highest entropy value.

#### 4.3.2 HSDC-X Algorithm

---

Input: Train\_vector  $\{Tr_{FHI}V\}$ , Test\_vector  $\{Te_{FHI}V\}$  and Labels

Output: Classification results

---

1. For all  $tev \in \{Te_{FHI}V\}$
  2. Size\_F = Sizeof( $Tr_{FHI}V$ )
  3. Fetch the class labels
  4. S = Sizeof( $Te_{FHI}V$ )
  5. End For
  6. For each  $tev(i, j)$ ,  $i = 1$  to  $m$ ,  $j = 1$  to  $n$
  7.  $Train_i = tev(i, j)$
  8.  $uniclass = distinct(Label)$
  9. L = Sizeof( $uniclass$ )
  10. End For
  11. If (Label is a unique class of  $iTrain$ )
  12. Class\_new = Label
  13. End if
  14. SVM struct = Svmtrain( $Tr_{FHI}V$ , Class\_new, 'svm kernel', 'rbf')
  22. Obtain  $Trl(X)$  and  $Trstl(X)$  from  $Tr_{FHI}V$  and  $Train_t$
  15. Train\_CNN(CL, Sub-sampling layer)
- 

(Continued)

---

Continued

---

16. Bz = batch size(tev)
  17. CNN\_result = test\_y(cnn\_output)
  18. Return → classification results (COVID-19, Pneumonia, Normal)
  19. Stop
- 

Each feature of the image is picked up from the test set of X-ray features. The dimension of the feature matrix is computed. Then the SVM Struct Uses the SVM kernel and RBF functions. Then the SVM training algorithm estimates the training features. In the next phase, CNN is trained, and the batch size of the testing feature is computed. The neurons from the CNN output layer are returned as the classification results as COVID-19, Pneumonia, and Normal.

---

**Pseudo code of HSDC-X algorithm**


---

1. Fetch the testing X-ray images  $T_e$
  2. Get the feature vectors  $FHI(V_j)$  of  $T_e$  Using Eq. (15)
  3. Compute the size of the training feature vector matrix  $tr$
  4. Read the class values of the  $tr$
  5. Compute the size of the testing vector matrix  $tev$
  6. Get the unique class with length  $N$  of each  $tev$
  7. Obtain the class  $Train\_T$  of each training feature vector  $trv$
  8. If Label is a unique class of training vector,
  9. Train each  $trv$  Using svm and rbf kernel function
  10. Obtain the training class of each  $trv$  and  $tev$  Using the SVM training algorithm
  11. Train CNN Using CL and Sub-sampling layer
  12. Compute the batch length of  $tev$
  13. Find out neurons from the output layer of CNN
  14. Return the classification results as {COVID-19, Pneumonia, Normal }
- 

## 5 Simulation Results

The proposed HSDC model was constructed in Python 8. The COVID-19 datasets for CT scan images are obtained from the Kaggle dataset [23]. The database of Wang et al. [24] was utilized for pneumonia and normal X-ray images. The accuracies of proposed classifiers are compared against DCNN, SVM, RF, K-Nearest Neighbor (KNN), AlexNet, VGG-16, and ResNet-50 classifiers by measuring the Accuracy, Precision, Recall, and F1-score metrics. Table 5 lists the settings used for HSDC classification.

### 5.1 Results of CT Scan Images

In the CT scan images, 213 severe cases, 236 moderate cases, and 167 normal cases are taken for input. Fig. 3 shows some of the input images from normal, moderate, and severe categories.

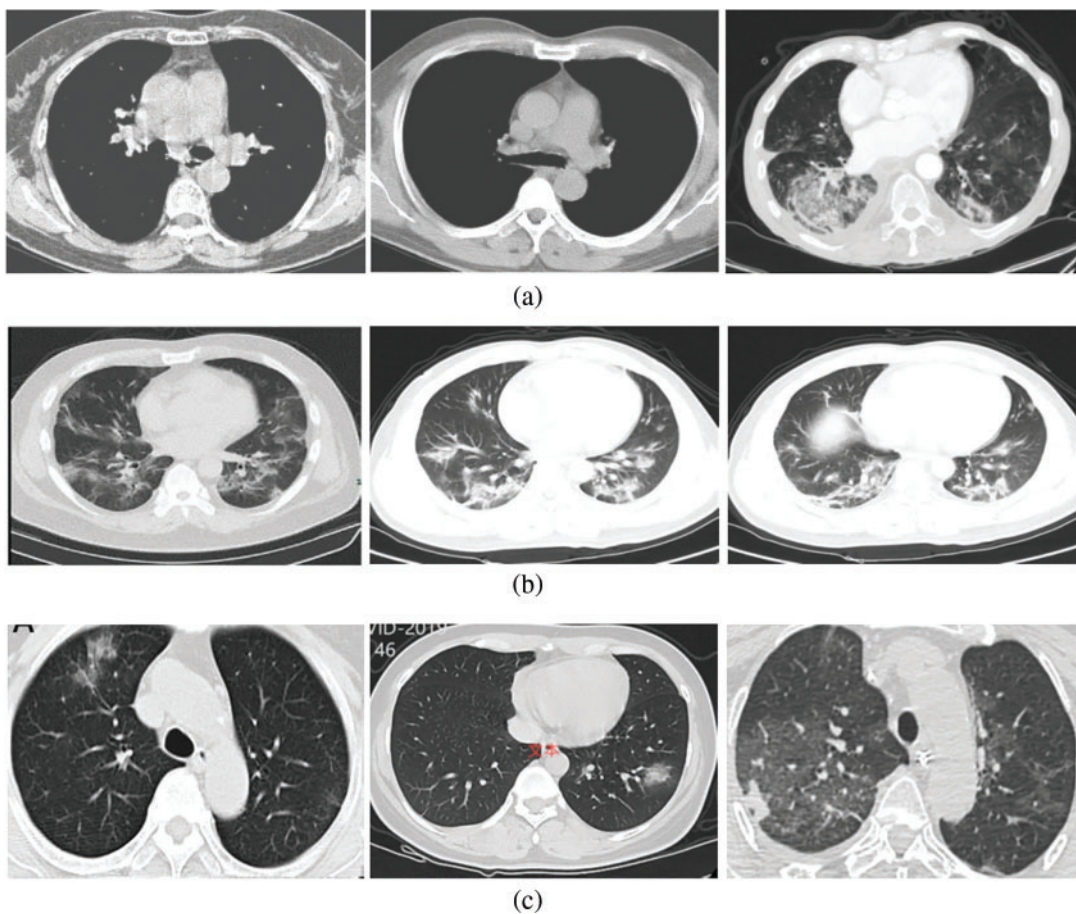
Table 6 shows each category's classified images and their accuracy values.

It is natural to generate accuracy curves for an ML model on training and validation (TV) images during training. The training curve computed from the training dataset provides details about the accuracy of the HSDC-CT. The validation curve determined from a validation dataset includes information on obtained model accuracy. The TV loss and accuracy curves of HSDC-CT for 80 epochs

are presented in Figs. 4a and 4b, respectively. The figure shows that the validation curve slightly differs from the training curve for higher epochs.

**Table 5:** HSDC settings





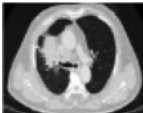

HSDC parameters	Values
Number of classes	3
Activation function	ReLU
Threshold value	0.3
Number of neurons in dense layers	32
FC	84
Learning rate	0.01

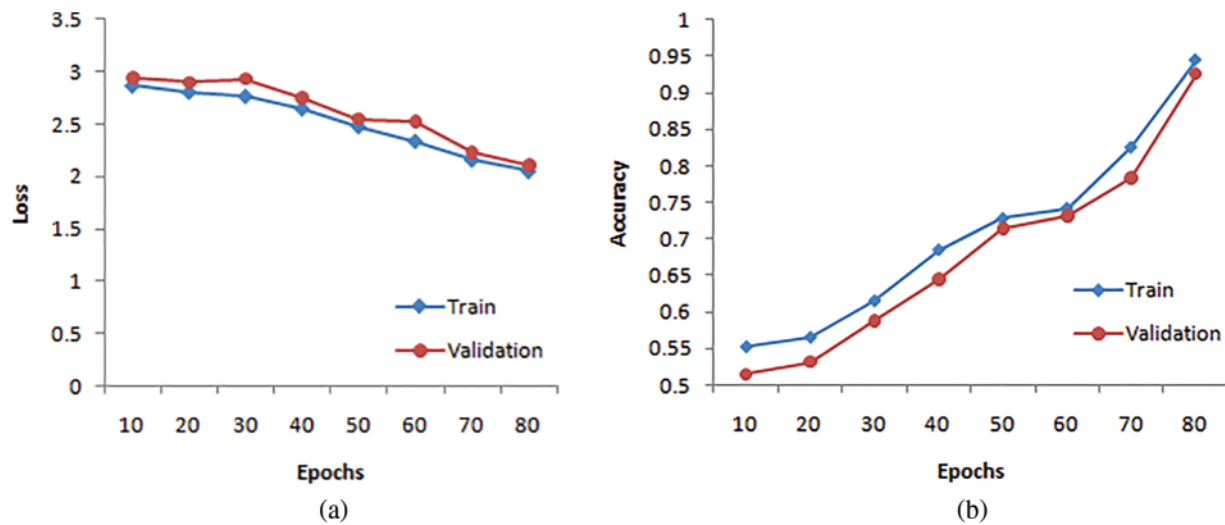


**Figure 3:** (a) Normal (No-COVID) cases (b) Moderate cases (c) Severe cases



**Table 6:** Classified images for each category

Input	Processed	Class	Accuracy	Precision	Recall
		Normal	94.63	91.56	88.67
		Moderate	92.72	90.67	89.46
		Severe	94.56	90.45	90.35



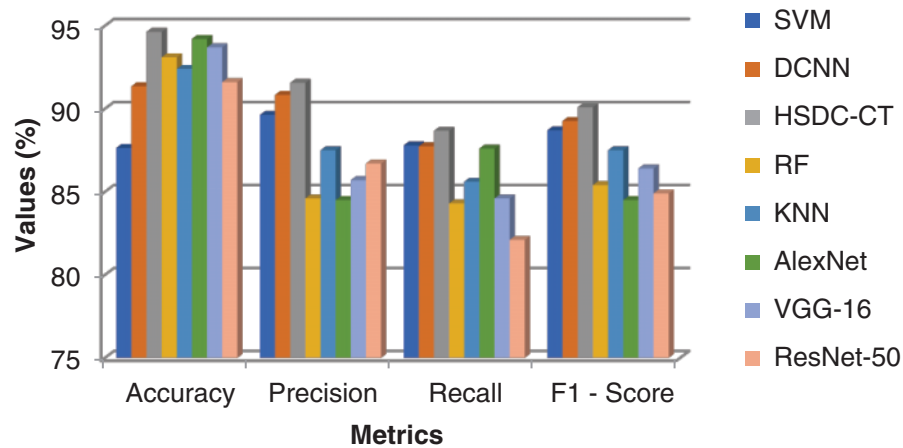
**Figure 4:** (a) Loss curves for HSDC-CT (b) Accuracy curves for HSDC-CT

Table 7 and Fig. 5 show the performance results for various classifiers.

**Table 7:** Results of all the classifiers for CT-images

Metrics	SVM	RF	KNN	AlexNet	VGG-16	ResNet-50	DCNN	HSDC-CT
Accuracy	87.6	93.1	92.4	94.2	93.7	91.6	91.3	94.6
Precision	89.6	84.6	87.5	84.5	85.7	86.7	90.8	91.5
Recall	87.8	84.3	85.6	87.6	84.6	82.1	87.7	88.6
F1-score	88.7	85.4	87.5	84.5	86.4	84.9	89.3	90.1

Fig. 5 shows that the accuracy of HSDC-CT is around 94.6, followed by other algorithms. Similarly, it outperforms the different techniques in terms of other metrics also.



**Figure 5:** Comparison results of all classifiers for CT images

### *Impact of Noise*

In the next experiment, the performance of HSDC-CT and DCNN classifiers is evaluated in the presence of noises. The input images are corrupted with Gaussian, salt, and pepper noises. [Table 8](#) shows the results for different noise values.

**Table 8:** Accuracy values for various noise ranges

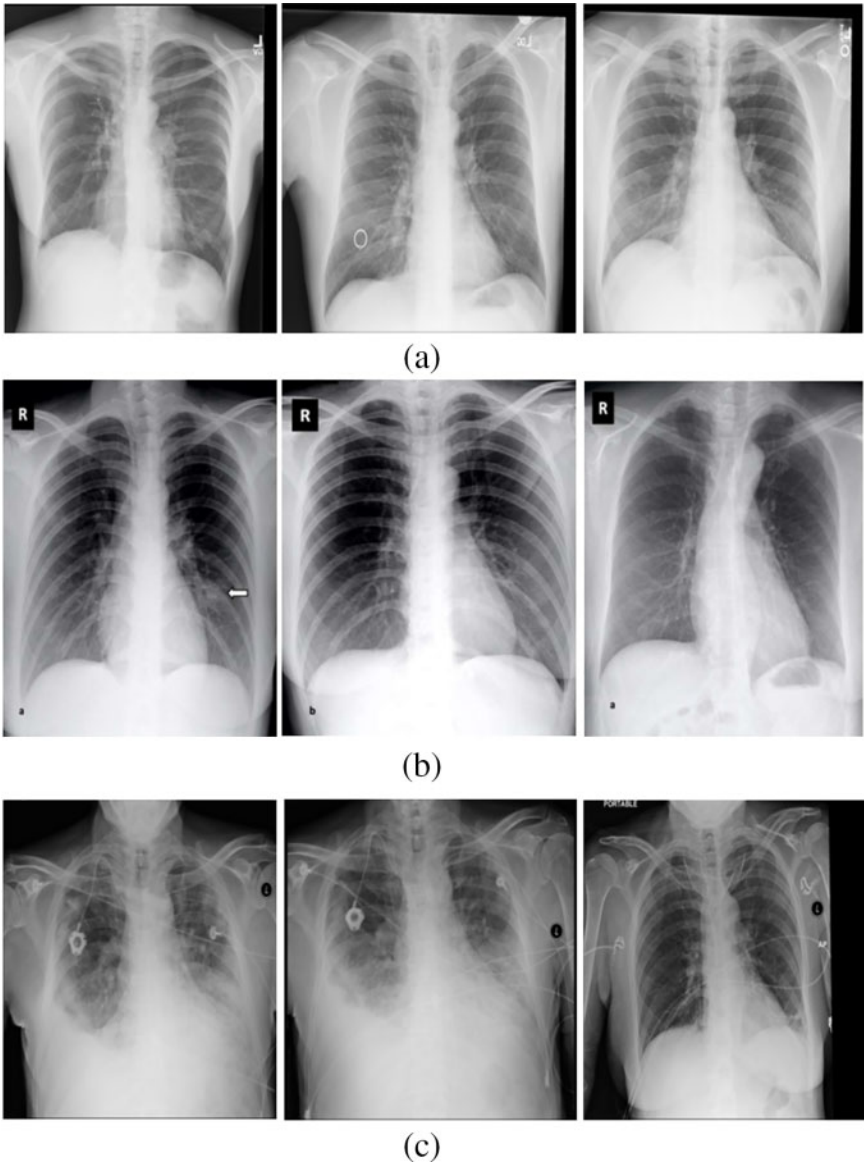
Noise type	Variance	Accuracy		Precision		Recall		F1-score	
		HSDC-CT	DCNN	HSDC-CT	DCNN	HSDC-CT	DCNN	HSDC-CT	DCNN
Gaussian	0.01	93.8	90.2	90.6	88.6	87.2	86.5	89.2	88.4
Salt & pepper	0.05	92.7	88.7	89.5	87.4	86.4	85.4	88.5	86.7
Gaussian	0.05	91.5	85.5	88.4	86.1	85.5	84.2	87.4	85.3
Salt & pepper	0.1	90.7	85.1	86.7	84.9	84.2	83.1	85.7	83.1

### *5.2 Results of X-Ray Images*

Around 234 normal, 390 Pneumonia, and 267 COVID-19 X-ray images are considered as input. [Fig. 6](#) contains the images of various categories. The preprocessing results of the MADF technique are demonstrated in [Fig. 7](#).

[Table 9](#) shows the classified X-ray images.

The TV Loss and accuracy curves of HSDC-X for 80 epochs are presented in [Figs. 8a](#) and [8b](#), respectively. The figure shows that the validation curve has little deviation from the training curve for higher epochs.



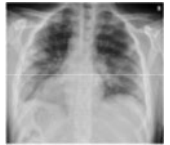
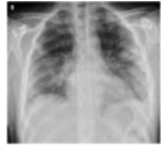




**Figure 6:** (a) Normal category (b) COVID-19 category (c) Pneumonia category



**Figure 7:** Preprocessed images

**Table 9:** Classification of X-ray images

Input	Classified	Category	Accuracy	Precision	Recall
		Normal	96.6	94.4	93.5
		Covid-19	96.4	94.7	92.4
		Pneumonia	95.2	92.3	91.7

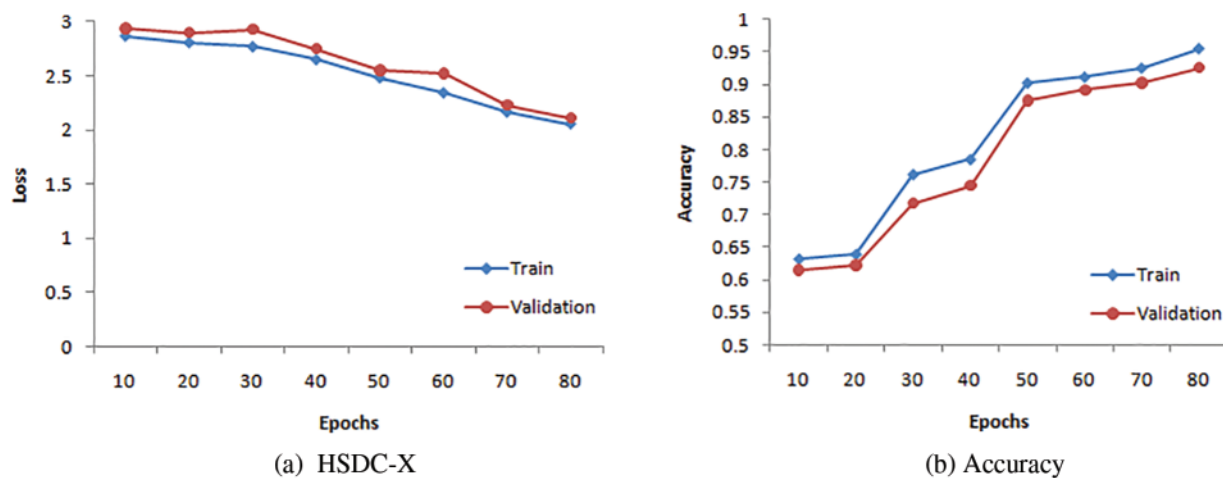
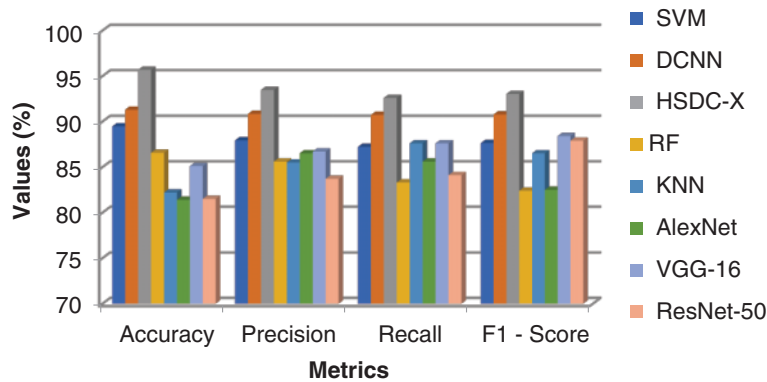
**Figure 8:** Training and validation curves for HSDC

Table 10 and Fig. 9 show the performance results of the algorithms.

**Table 10:** Comparison results of all the approaches for X-ray images

Metrics	SVM	RF	KNN	Alex Net	VGG-16	ResNet-50	DCNN	HSDC-X
Accuracy	89.46	86.56	82.2	81.4	85.1	81.5	91.29	95.67
Precision	87.93	85.6	85.5	86.5	86.7	83.7	90.84	93.47
Recall	87.21	83.3	87.6	85.6	87.6	84.1	90.72	92.59
F1-score	87.63	82.4	86.5	82.5	88.4	87.9	90.77	93.02



**Figure 9:** Comparison results of all the techniques for X-ray images

Since HSDC-X follows the training procedures of both SVM and CNN, it attains higher accuracy than the other two approaches. Fig. 9 shows that the accuracy of HSDC-X is around 95.6, followed by other algorithms. Similarly, it outperforms the different techniques in terms of other metrics also.

### 5.3 Convergence Time (CT) and Computational Complexity (CC)

The convergence time and computational complexities of the new and existing algorithms were analyzed on GPU settings, as shown in Table 11.

**Table 11:** CT and CC of various algorithms

Model	Complexity (in seconds)	Convergence time (seconds)
HSDC	11.25	0.010
SVM	10.45	0.014
CNN	10.30	0.021

As depicted in Table 11, the proposed HSDC model has the least convergence time, around 0.010 s which is lesser than SVM and CNN. But its computational complexity is slightly higher, around 11.25 s, than SVM and CNN, due to the integration of SVM and CNN.

#### Impact of Noise

In the next experiment, the performance of HSDC-X and DCNN classifiers is evaluated in the presence of noises. Table 12 shows the results for different noise values.

**Table 12:** Accuracy values for various noise ranges

Noise type	Variance range	Accuracy		Precision		Recall		F1-score	
		HSDC-X	DCNN	HSDC-X	DCNN	HSDC-X	DCNN	HSDC-X	DCNN
Gaussian	0.01	94.2	90.5	92.6	89.8	91.5	89.5	92.6	90.4
Salt & pepper	0.05	93.7	89.7	91.5	89.4	90.6	88.4	91.5	88.7
Gaussian	0.05	93.6	88.4	91.3	88.7	90.5	87.6	90.8	89.5
Salt & pepper	0.1	92.2	87.5	90.4	88.1	88.6	87.2	90.2	86.2

## 6 Conclusions

In this paper, the HSDC model was proposed for COVID-19 detection and classification. For CT-scan Images, the histogram threshold is computed using the ICHSeg algorithm. The selected regions from the feature vector are classified using the HSDC-CT algorithm as normal, moderate, and severe. For X-ray images, the FHI features are used in the training process. The X-ray images are categorized Using the HSDC-X algorithm. The HSDC model has been implemented in Python. Experimental results have shown that the accuracy of the HSDC model attains the highest of 94.6 for CT scan images and 95.6 for X-ray images, than the other algorithms.

Limitations: Though this work predicts the severity of COVID-19 infection only from images, it could not be able to predict the death and survival rate of the patients.

Future work focuses on predicting the survival and death rate of COVID-19 patients from their case history database.

**Funding Statement:** The authors received no specific funding for this study

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] L. Brunese, F. Martinelli, F. Mercaldo and A. Santone, "Machine learning for coronavirus covid-19 detection from chest X-rays," *Procedia Computer Science*, vol. 176, pp. 2212–2221, 2020.
- [2] M. M. Rahaman, C. Li, Y. Yao, F. K. M. A. Rahman *et al.*, "Identification of COVID-19 samples from chest X-ray images using deep learning: A comparison of transfer learning approaches," *Journal of X-Ray Science and Technology*, vol. 28, pp. 821–839, 2020. <https://doi.org/10.3233/XST-200715>
- [3] N. S. Shadin, S. Sanjana and N. J. Lisa, "COVID-19 diagnosis from chest X-ray images using convolutional neural network (CNN) and inceptionv3," in *Int. Conf. on Information Technology (ICIT)*, Amman, Jordan, 2021.
- [4] B. Sekeroglu and I. Ozsahin, "Detection of COVID-19 from chest X-ray images using convolutional neural networks," *SLAS Technology*, vol. 25, no. 6, pp. 553–565, 2020.
- [5] S. Pradeep, C. Malathy, T. Vardhan and T. Sainadh, "Detection of malware from IoT devices using deep learning techniques," *Journal of Physics: Conference Series*, 2021. <https://doi.org/10.1088/1742-6596/1818/1/012219>
- [6] R. Jayaram and T. Senthil Kumar, "Onboard cloud-enabled Parkinson disease identification system using enhanced grey wolf optimization," *Computing Technology and Information Management*, vol. 18, pp. 1360–1372, 2021.
- [7] D. A. Garzón, J. A. Grisales, S. O. Arias, H. B. Ar Arteaga, M. A. Bravo-Ortiz *et al.*, "COVID-19 detection in X-ray images using convolutional neural networks," *Machine Learning with Applications*, vol. 6, no. 100138, 2021. <https://doi.org/10.1016/j.mlwa.2021.100138>
- [8] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim *et al.*, "Automated detection of COVID-19 cases using deep neural networks with X-ray images," *Computers in Biology and Medicine*, vol. 121, no. 103792, 2020. <https://doi.org/10.1016/j.compbiomed.2020.103792>
- [9] M. Alotaibi and B. Alotaibi, "Detection of COVID-19 using ddeep learning on X-ray images," *Intelligent Automation & Soft Computing*, 2021. <https://doi.org/10.32604/iasc.2021.018350>
- [10] M. Aminu, N. A. Ahmad and M. H. M. Noor, "COVID-19 detection via deep neural network and occlusion sensitivity maps," *Alexandria Engineering Journal*, vol. 2021, no. 60, pp. 4829–4855, 2021.
- [11] R. Jain, M. Gupta, S. Taneja and D. Jude Hemanth, "Deep learning based detection and analysis of COVID-19 on chest X-ray images," *Applied Intelligence*, vol. 51, pp. 1690–1700, 2021.

- [12] Y. Qiblaweya, A. Tahira, E. H. Muhammad, H. Chowdhurya, A. Khandakara *et al.*, “Detection and severity classification of COVID-19 in CT images using deep learning,” *MDPI*, vol. 11, no. 5, 2021. <https://doi.org/10.3390/diagnostics11050893>
- [13] D. Müller, I. S. Rey and F. Kramer, “Robust chest CT image segmentation of COVID-19 lung infection based on limited data,” *Informatics in Medicine*, vol. 25, no. 100681, 2021. <https://doi.org/10.1016/j.imu.2021.100681>
- [14] M. O. Arowolo, R. O. Ogundokun, S. Misra, F. Kadri and T. O. Aduragba, “Machine learning approach using KPCA-SVMs for predicting COVID-19,” *Healthcare Informatics for Fighting COVID-19 and Future Epidemics*, vol. 1, pp. 193–209, 2022.
- [15] M. O. Arowolo, R. O. Ogundokun, S. Misra, B. D. Agboola and B. Gupta, “Machine learning-based IoT system for COVID-19 epidemics,” *Computing*, vol. 1, pp. 831–847, 2022.
- [16] M. O. Arowolo, M. O. Adebisi, E. P. Michael, H. Eric Aigbogun, S. O. Abdulsalam *et al.*, “Detection of COVID-19 from chest X-ray images using CNN and ANN approach,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 6, pp. 754–759, 2022.
- [17] C. Huang, X. Li and Y. Wen, “AN OTSU image segmentation based on fruitfly optimization algorithm,” *Alexandria Engineering Journal*, vol. 60, pp. 183–188, 2021.
- [18] J. Wang, Z. Cheng, O. K. Ersoy, M. Zhang, K. Sun *et al.*, “Improvement and application of chicken swarm optimization for constrained optimization,” *IEEE Access*, vol. 7, pp. 58053–58072, 2019.
- [19] N. Alam, M. Ahsan, M. A. Based, J. Haider and M. Kowalski, “COVID-19 detection from chest X-ray images using feature fusion and deep learning,” *Sensors, MDPI*, vol. 21, no. 4, 2021. <https://doi.org/10.3390/s21041480>
- [20] N. Khehrah, M. S. Farid, S. Bilal and M. H. Khan, “Lung nodule detection in CT images using statistical and shape-based features,” *Journal of Imaging, MDPI*, vol. 6, no. 2, 2020. <https://doi.org/10.3390/jimaging6020006>
- [21] M. J. J. Ghrabat, G. Ma, I. Y. Maolood, S. S. Alresheedi and Z. A. Abduljabbar, “An effective image retrieval based on optimized genetic algorithm utilized a novel SVM-based convolutional neural network classifier,” *Human Centric Computing Information Sciences*, vol. 9, no. 3, 2019. <https://doi.org/10.1186/s13673-019-0191-8>
- [22] M. Baskar, J. Ramkumar, R. Rathore and R. Kabra, “A deep learning based approach for automatic detection of bike riders with no helmet and number plate recognition,” *International Journal of Advanced Science and Technology*, vol. 29, no. 4, pp. 1844–1854, 2020.
- [23] 2020. <https://www.kaggle.com/c/covidct/data?select=Train>
- [24] 2017. [https://Openaccess.thecvf.com/Content\\_cvpr\\_2017/Papers/Wang\\_ChestX-ray](https://Openaccess.thecvf.com/Content_cvpr_2017/Papers/Wang_ChestX-ray)