

Modeling of Archimedes Optimization Algorithm with Deep Learning Driven Automated Social Distancing Detection and Classification

Dhanya G Nair¹, Dr.K.P.SanalKumar², Dr.S.AnuHNair³

¹Research Scholar, Department of Computer and Information Sciences, Annamalai University, Chidambaram, India.

²Assistant Professor, PG Department of Computer Science, R.V Government Arts College, Chengalpattu, India.

³Assistant Professor, Department of CSE, Annamalai University, Chidambaram, India [Deputed to WPT, Chennai].

Abstract

Social distancing (SD) detection refers to the usage of technology, especially image processing and computer vision techniques, to monitor and identify compliance with SD guidelines in public spaces. The objective is to automatically analyze images or video feeds to identify individuals and measure the distances between them. Leveraging convolutional neural networks (CNN) or similar deep learning (DL) methods, the system analyzes images or video feeds to identify the presence of individuals and measure the distances between them. By learning sophisticated spatial patterns and relationships, the DL algorithm identifies instances where safe distance is not maintained, which provides real-time visualizations or alerts to assist in imposing public health measures. This study develops an Archimedes Optimization Algorithm with Deep Learning Driven Automated Social Distancing Detection and Classification (AOADL-SDDC) technique. The purpose of the AOADL-SDDC technique is to exploit hyperparameter tuned DL model for SD recognition. Primarily, the AOADL-SDDC technique makes use of Wiener Filter (WF) based noise removal and Dynamic Histogram Equalization (DHE) based contrast enhancement. For pedestrian detection process, the UNet++ with RMSProp optimizer is used, which detects the pedestrians accurately and the distance among them can be computed by the use of Manhattan distance. To detect SD, Bi-directional convolutional LSTM (BiConvLSTM) model can be employed and its hyperparameters can be adjusted by the use of AOA. A comprehensive set of experiments was carried out for examining the performance of the AOADL-SDDC technique. The stimulation value stated that the AOADL-SDDC technique reaches superior performance over other models.

Keywords: Social Distancing Detection; Deep Learning; Parameter Tuning; Dynamic Histogram Equalization; Segmentation

1. Introduction

Social distancing (SD) is one of the effective public health actions that reduce virus spread by decreasing physical interaction among people [1]. SD measure is definite and executed in numerous methods at dissimilar stages based on country. WHO has explained rules and regulations for SD action based on circumstances and places like factories, universities, malls, airports, and so on [2]. A European Centre for Display Prevention and Control (ECDC) well-defined SD based on specific and cluster levels. The SD plans differ so there are many monitoring systems obtainable [3]. It is mainly supervised by non-profit governments and understood as occurrence or grade of execution of plan in dissimilar states [4]. Computer vision (CV) systems feature non-invasive data gathering and nonstop observing, which procedures for handling composed videos and images are central modules [5].

Usually, handling techniques are generated by employing image processing (IP), machine learning (ML) techniques, or combination of both [6].

IP techniques are very simple to execute but subject to image or video with unpredictable backgrounds and illuminations, flexible animal sizes and shapes, similarity amid animals, animal obstruction, and overlapping as well as lowest resolutions [7]. Artificial intelligence (AI) techniques replicate the way humans increase certain sorts of data and are most prevalent models for computer vision-based animal computation. ML methodologies are a subset of AI that normally involves physically intended feature extractors for additional detection as well as for enhancing portability and sturdiness for estimate [8]. However, creating feature extractors needs lengthy alteration and practical knowledge, which bounds huge apps of ML-based computer vision systems in SD. DL models established from ML are representation-learning techniques and can mechanically find data representations or features from raw data deprived of wide physical feature extraction [9]. The models have been gradually proposed in IP from SD because they can enhance exactness and portability for behavior detection, thus resolving the above-mentioned problems of physical labeling and automatically determining space behavior [10].

This study develops an Archimedes Optimization Algorithm with Deep Learning Driven Automated Social Distancing Detection and Classification (AOADL-SDDC) technique. The purpose of the AOADL-SDDC technique is to exploit hyperparameter tuned DL model for SD recognition. Primarily, the AOADL-SDDC technique makes use of Wiener Filter (WF) based noise removal and Dynamic Histogram Equalization (DHE) based contrast enhancement. For pedestrian detection process, the UNet++ with RMSProp optimizer is used, which detects the pedestrians accurately and the distance among them can be computed by the use of Manhattan distance. To detect SD, Bi-directional convolutional LSTM (BiConvLSTM) model can be employed and its hyperparameters can be adjusted by the use of AOA. A detailed set of experiments was conducted to examine the performance of AOADL-SDDC technique.

2. Literature works

Dave et al. [11] concentrate on watching the individuals who keep social distancing from the video by employing a pre-trained YOLOv4 procedure. This method is used to identify manifold humans and estimate the distance among dual objects, and if two humans are against the rules and regulations of SD, then an alert will be generated. Besides, predefined YOLOv4 techniques are equated with projected models. Walia et al. [12] developed a real-time surveillance method. The projected method includes enchanting input t the CCTV feed and identifying people in the frame, by employing YOLOv5 method. Then, these noticed aspects are handled by employing Stacked ResNet-50 for identification of whether the individual can wear face mask or not. The DBSCAN was applied in order to discover closeness in people noticed.

Hermina et al. [13] concentrate on defining the SD in educational institutions primarily in colleges. The people are perceived from video clips by utilizing YOLOV3 model which is trained on a dataset of Common Objects in Context (COCO). Then safe distance between people is found by utilizing a Euclidean distance. The minimum threshold value is stable to the safe space recommended by WHO. Oztel et al. [14] developed an automatic tactic for identifying SD as well as face masks. For this method, a two-cascaded YOLO is utilized. In addition, a dual-part feature extraction technique employed with YOLO has been projected. An initial portion of developed feature extraction model removes common features utilizing the TL method. The next portion removes superior features exact to this present challenge by utilizing the layers of classification and LBP. Shete [15] concentrates and goals its research near executing an SD and Face Mask Recognition Method. This will execute object as well as facial detection in video streams of walkers. Pertained techniques like YOLOv3, ResNet Classifier, and DSFD have been employed. People who are not at a minimum distance and faces without masks were discovered. In [16], a DL-assisted drone is considered for face mask recognition as well as SD observing. This system chiefly efforts on Industrial-IoT (IIoT) observing by employing Raspberry-Pi 4. This drone automation method will send alarms to persons through a speaker to uphold SD. In addition, it takes images and spots uncovered people by employing a faster R-CNN technique.

3. The Proposed Model

In this study, we develop an innovative AOADL-SDDC system. The purpose of the AOADL-SDDC technique is to exploit hyperparameter tuned DL model for SD recognition. The AOADL-SDDC technique comprises three major processes namely pre-processing, segmentation, and classification. At the pre-processing stage, the quality of the input images can be enhanced by the use of WF based noise removal and DHE based contrast improvement. Besides, the segmentation process takes place where the UNet++ with RMSProp optimizer and Manhattan distance based pedestrian distance calculation are involved. Finally, the AOA with BiConvLSTM model can be exploited for accurate detection and classification of the SD. Fig. 1 represents the entire procedure of AOADL-SDDC methodology.

3.1. Image Pre-processing

In the initial stage, the WF based pre-processing and DHE based contrast enhancement process is performed to boost the quality of the input images.

3.1.1. WF based Pre-processing

The WF based pre-processing is utilized to eliminate the noises at the input images. In image processing, the WF excels as a powerful mechanism for noise removal [17]. Tailored for scenarios where the signal and noise features are known, the WF exploits a frequency-domain method for minimizing the noise while retaining the underlying signal. In pedestrian detection applications, where image quality is predominant, the WF proved to be advantageous for improving the overall accuracy, reducing unwanted artifacts, and enhancing the clarity of features.

3.1.2. DHE based Contrast Enhancement

The DHE approach is exploited for contrast enhancement in the images. DHE is an effective tool for enhancing contrast, especially while handling images with different illumination conditions [18]. Different from static histogram equalization, DHE adapted to local variation in intensity, which ensures optimum contrast across dissimilar regions of image. Applied to pedestrian detection, where scene presents different lighting scenarios, DHE contributes to enhanced visibility of pedestrian and their surroundings. DHE has become paramount in preparing images for subsequent processing by dynamically adjusting the histogram to enhance local contrast, which enhances the overall efficiency of pedestrian detection algorithm and facilitates accurate feature extraction.

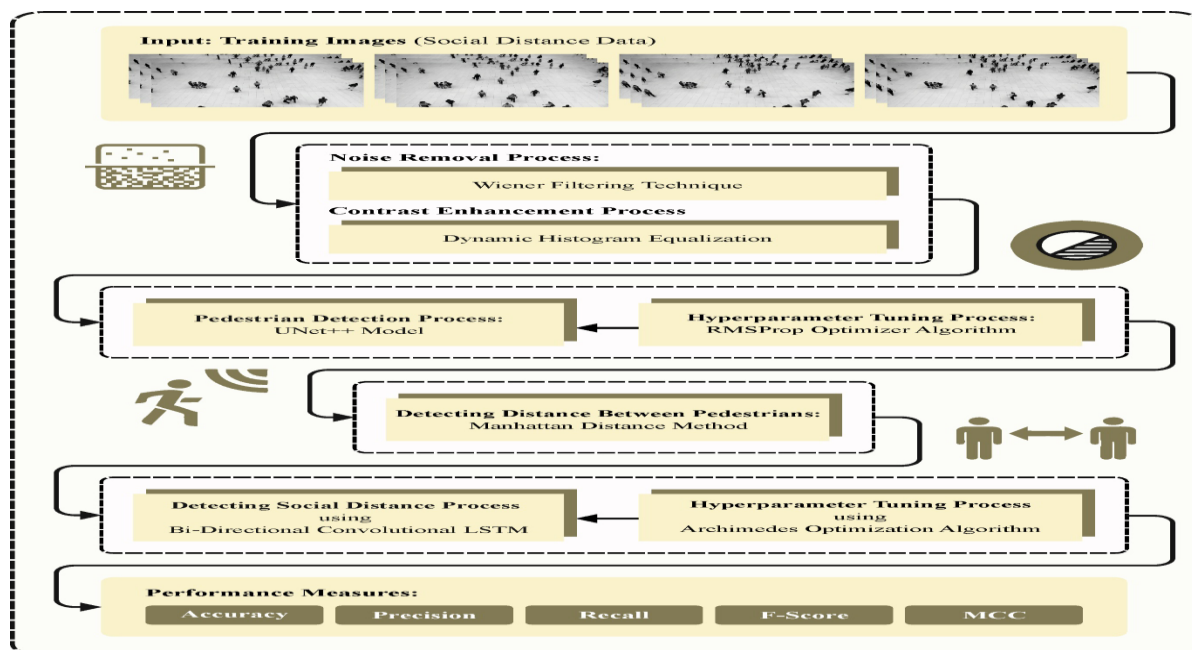


Fig. 1. Overall procedure of AOADL-SDDC method

3.2. Image Segmentation

Once the input images are pre-processed, the next stage is to perform segmentation to detect the occurrence of pedestrians and estimate the distance between them using Manhattan distance.

3.2.1. UNet++ With RMSProp Optimizer

UNet++ becomes a formidable choice for pedestrian detection, leveraging its advanced nested skip pathway structure to capture multi-scale contextual information and enhance feature extraction [19]. In terms of pedestrian detection, the UNet++ structure excels at handling complex spatial relationships and diverse appearances within crowded scenes. The nested skip connection, intricately weaving features from dissimilar scales, enables one to precisely describe pedestrians against different backgrounds, making it fit for the nuanced demands of these applications. UNet++'s capability to capture hierarchical features plays a major role in accomplishing accurate segmentation, essential for succeeding in tracking and recognizing pedestrians.

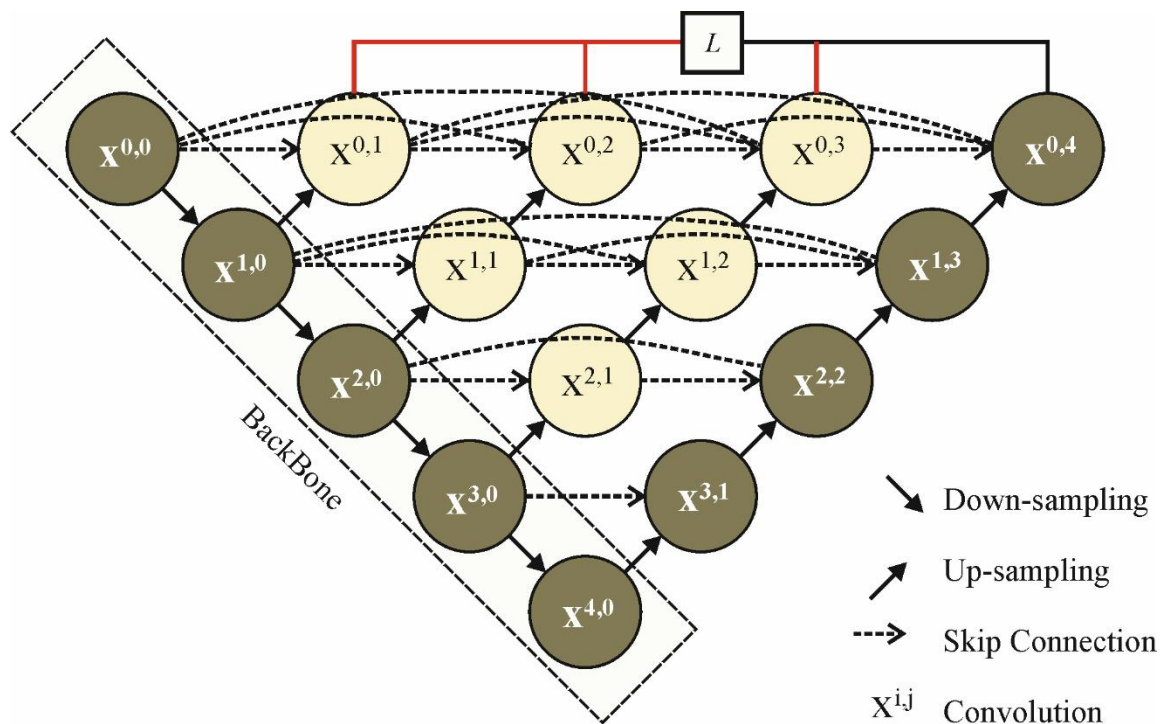


Fig. 2. Architecture of UNet++

While applying UNet++ for pedestrian detection, the selection of optimizer plays a pivotal role in effective model training. Fig. 2 depicts the architecture of UNet++. The RMSProp optimizer, with its adaptive learning rate algorithm, proves advantageous in alleviating problems related to DNN. Particularly, the visual features of pedestrians might differ considerably across dissimilar scenes, RMSProp adapts the learning rate for all the parameters based on the historical gradient data. This adaptability allows more stable convergence, which prevents problems such as exploding gradients or vanishing. The UNet++ model trained with RMSProp efficiently capture intricate feature and perform in a stable and robust manner, which improves the overall reliability and accuracy of pedestrian detection in different environments.

3.2.2. Distance Calculation Using Manhattan Distance

To compute the distance between the detected pedestrians, the Manhattan distance based distance calculation is applied. Manhattan distance, also termed as city block distance or L1 distance, is a measure of distance between two points in a grid-based system measured along the grid line [20]. It is frequently utilized in computer vision and image processing for tasks like object detection.

To utilize Manhattan distance for pedestrian detection, it usually represents the location of object or pedestrian as point in the grid, and then evaluates the Manhattan distance between the points:

Represent Object Position: the pedestrian position or object in the scene is represented as point in a grid. Every point represents a coordinate pair (x,y), where x and y are the horizontal and vertical positions in the grid.

Calculate Manhattan Distance: Assume a 2 points A(x1, y1) and B(x2, y2), the Manhattan distance (L1 distance) can be evaluated by:

$$\text{Manhattan distance} = |x_2 - x_1| + |y_2 - y_1| \quad (1)$$

Set Threshold for Detection: threshold distance defines when the pedestrian is regarded as detected. When the evaluated Manhattan distance between the pedestrian and a reference point is below the threshold, then pedestrian is considered as detected.

3.3. BiConvLSTM based Classification

For accurate detection of SD, the BiConvLSTM model is used. As the most basic RNN architecture, the RNN unit has clear structure and the data transmission mode is given below [21]:

$$h_t = \tanh(W_h[h_{t-1}, x_t] + b_h), \quad (2)$$

In Eq. (1), the corresponding weights and biases are W_h and b_h . The output h_t hidden state at time t is collectively defined by the input data x_t and the h_{t-1} hidden state at $t - 1$ time. The LSTM model presents three gating mechanisms for controlling the data flow and exploits a cell state to store the prior data, thus resolving the exploding or vanishing gradient problems.

The internal architecture of LSTM has extremely complicated, and the computation process is expressed in the following. The mathematical model of its data propagation is given below:

$$\Gamma_f^{<t>} = \sigma(W_f[h_{t-1}, x_t] + b_f), \quad (3)$$

$$\Gamma_i^{<t>} = \sigma(W_i[h_{t-1}, x_t] + b_i), \quad (4)$$

$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c), \quad (5)$$

$$c_t = \Gamma_f^{<t>} * c_{t-1} + \Gamma_i^{<t>} * \tilde{c}_t \quad (6)$$

$$\Gamma_o^{<t>} = \sigma(W_o[h_{t-1}, x_t] + b_o), \quad (7)$$

$$h_t = \Gamma_o^{<t>} * \tanh(c_t), \quad (8)$$

Where h_t shows the hidden state at t time produced by the output gate. Γ_f , Γ_i and Γ_o are the forget, input, and output gates that differ over time, $\tilde{c}_t$ indicates the candidate cell state calculated from the prior hidden state h_{t-1} and the novel input x_t , c_t shows the cell state updated by integrating $\tilde{c}_t$ and the prior cell state c_{t-1} .

BiConvLSTM (Bi-directional convolutional LSTM) cell is used for accessing the long-range context to combine the extracted hierarchical feature from the recursive inference blocks [22]. The input, which is the concatenation of recursive inference block output based on the sequence, is represented by $[F_n, F_{n-1}, \dots, F_1]$. ConvLSTM and BiConvLSTM are used for learning global, long-term spatiotemporal features of video. In this work, the recursion of RIB is considered a temporal series.

(1) BiConvLSTM layer: A BiConvLSTM cell has multiple ConvLSTM cells with two cell states, separately a forward and backward sequence cell, to model the sequence dependency from the past and present recursions, and access long-term dependency feature in both directions of recursion series. The formula of backward or forward ConvLSTM cell is given as follows:

$$i_n^f = \sigma(W_{fi}^f * F_n + w_{hi}^f * H_{n-1}^f + b_i^f) \quad (9)$$

$$f_n = \sigma(W_{ff}F_n + W_{hf} * H_{n-1} + b_f) \quad (10)$$

$$o_n = \sigma(W_{fo} * F_n + W_{ho} * H_{n-1} + b_o) \quad (11)$$

$$C_n = f_n \odot C_{n-1} + i_n * \tanh(W_{fc} * F_n + W_{hc} * H_{n-1} + b_c) \quad (12)$$

$$H_n^f = o_n \odot \tanh(C_n) \quad (13)$$

Here " $\odot$ " shows the Hadamard product, " $*$ " indicates the convolution function; and " σ " represents the sigmoid function. W_{f*}^f and W_{h*}^f are the convolution kernel for the input and hidden stages in forward ConvLSTM cell, correspondingly.

(2) Multi-BiConvLSTM layer: ConvLSTM layer of backward sequence hidden and cell states as (h_b^i, c_b^i) and the forward sequence is hidden and cell states are represented as (h_f^i, c_f^i) . In the first layer, we concatenate the forward and backward hidden states and pass them through 3×3 Conv layer to obtain the hidden representation as input of next BiConvLSTM layer. Next, we concatenate the hidden representation of BiConvLSTM layer for the input of upscaling layer.

3.4. AOA based Parameter Tuning

Finally, the AOA algorithm adjusts the parameters related to the BiConvLSTM model. Hashim et al. introduced AOA a recent and powerful algorithm that drew their inspiration from the principle state of Archimedes [23]. The study shows that AOA was the potential for resolving optimizer problems and get near optimum problems at a shorter time. The mathematical formula of AOA has different steps:

Step 1: Initialization, a random number of individuals are created and stored in their location as given below:

$$O_i = lb_i + rand \times (ub_i - lb_i) \quad (14)$$

In Eq. (14), O_i is the i^{th} agent's location, ub_i and lb_i are the upper and lower bounds of i^{th} agents and $rand$ denotes the random vector within $[0,1]$.

$$den_i = rand$$

$$vol_i = rand \quad (15)$$

In Eq. (15), vol_i is the volume of i^{th} agent, and den_i is the density.

$$acc_i = lb_i + rand \times (ub_i - lb_i) \quad (16)$$

In Eq. (16), acc_i is the acceleration, ub_i and lb_i are the upper and lower bounds of i^{th} agent correspondingly.

Step 2: Density and Volume, using Eq. (17), volume and density must be updated

$$\begin{aligned} den_i^{t+1} &= den_i^t + rand \times (den_{best} - den_i^t) \\ vol_i^{t+1} &= vol_i^t + rand \times (vol_{best} - vol_i^t) \end{aligned} \quad (17)$$

In Eq. (17), vol_i^t and den_i^t are the volume and density of j^{th} agents at t point and vol_{best} and den_{best} denotes the best-yet volume and density at t point.

Step 3: Density Factor and Transfer Operator, collision among each other and agent take place, and then the agent attempts to reach equilibrium state. The transfer operator TF move between exploration and exploitation, as follows:

$$TF = \exp\left(\frac{t - t_{max}}{t_{max}}\right) \quad (18)$$

In Eq. (18), t and t_{max} are the present and maximal iteration counter. In addition, a reduction of density factor was included to get a near optimal solution (best):

$$d^{t+1} = \exp\left(\frac{t_{max} - t}{t_{max}}\right) - \left(\frac{t}{t_{max}}\right) \quad (19)$$

Step 4: Exploration, collision of individual takes place. If TF is less than 0.5, then the random material is selected and the agent i 's acceleration will be updated as follows:

$$acc_i^{t+1} = \frac{den_{mr} + vol_{mr} \times acc_{mr}}{den_i \times vol_i^{t+1}} \quad (20)$$

In Eq. (20), a random material's volume, acceleration, and density are vol_{mr} , acc_{mr} , and den_{mr} and the volume, acceleration, and density of i^{th} agent is vol_i , acc_i and den_i .

Step 5: Exploitation, no individual collisions occur. If the TF is higher than 0.5, then agent i 's acceleration will be updated as follows:

$$acc^{t+1} = \frac{den_{best} + vol_{best} \times acc_{best}}{den_i \times vol_i^{t+1}} \quad (21)$$

In Eq. (21), the best individual acceleration, density, and volume are acc_{best} , den_{best} , and vol_{best} correspondingly.

Step 6: Normalize Acceleration, acceleration can be normalized using the following expression:

$$acc_{i-norm}^{t+1} = u \times \frac{acc_i^{t+1} + \min(acc)}{\max(acc) - \min(acc)} + l \quad (22)$$

In Eq. (22), l and u are the lower and upper boundaries of normalization set to 0.1 and 0.9 correspondingly. acc_{i-norm}^{t+1} denotes the percentage of changing steps of all the individuals, and $\max(acc)$, and $\min(acc)$ is the maximal and minimal acceleration correspondingly.

Step 7: Updating position, Eq. (21) update the distinct position if TF is lesser than 0.5, or else, Eq. (24) is applied.

$$x_i^{t+1} = x_i^t + c_1 \times rand \times acc_{i-norm}^{t+1} \times d \times (x_{rand} - x_i^t) \quad (23)$$

$$x_{best}^{t+1} = x_{best}^t + F \times c_2 \times rand \times acc_{i-norm}^{t+1} \times d \times (T \times x_{best} - x_i^t) \quad (24)$$

Where x_i^t and x_{best}^t are i^{th} and best agent at t^{th} iteration, T is time function and equals $c_3 \times TF$ where c_3 falling within $[c_3 \times 0.3, 1]$ and takes a fixed ratio in the optimum position. d denotes the dimensionality, the constants c_1 and c_2 are equal to 2 and 6. Next, it gradually decreased to distance between the optimum and existing locations, and F refers to the flag motion direction, p denotes the probability and is evaluated as follows:

$$F = \{+1 \text{ if } p < 0.5 \text{ } -1 \text{ if } p > 0.5 \quad (25)$$

The fitness selection is the significant factor which influences the performance of AOA technique. The hyperparameter selection approach includes the solution encoding process to estimate the efficiency of candidate solutions. Here, the AOA approach considers accuracy as the primary criterion for designing the FF.

$$Fitness = \max(P) \quad (26)$$

$$P = \frac{TP}{TP + FP} \quad (27)$$

Where TP and FP are the true positive and the false positive values.

4. Result analysis

The SD detection analysis of the AOADL-SDDC system can be examined by employing the SD dataset [24]. The dataset contains 1000 instances with two class labels as described in Table 1.

Table 1 Details on dataset

Classes	No. of Instances
High_Risk	500

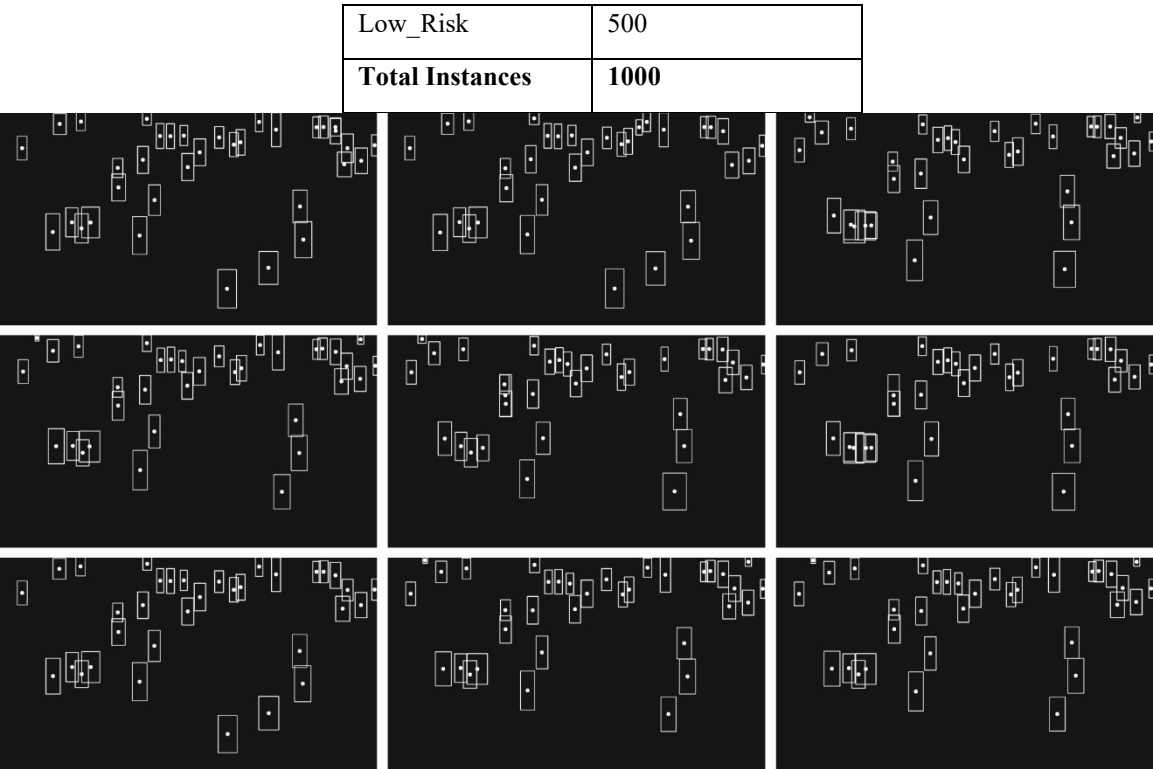


Fig. 3. Masked Regions



Fig. 4. Detected Regions

Fig. 3 displays the masked regions accomplished by the AOADL-SDDC technique and the corresponding SD detected fields are presented in Fig. 4. These acquired outcomes confirms that the AOADL-SDDC system precisely recognized the SD classes.

Fig. 5 displays the classifier analysis of the AOADL-SDDC system on 80:20 of training phase (TRPH)/testing phase (TSPH). Figs. 5a-5b illustrates the confusion matrix accomplished by the AOADL-SDDC technique. This figure signified that the AOADL-SDDC algorithm can be correctly identified and categorized with high_risk and low_risk classes. Additionally, Fig. 5c shows the PR analysis of the AOADL-SDDC method. The figure indicates the AOADL-SDDC technique gets higher PR performance with each class. Also, Fig. 5d represents the ROC analysis of the AOADL-SDDC algorithm. The figure exposed the AOADL-SDDC methodology provides effective outcomes with increased ROC values with diverse class labels.

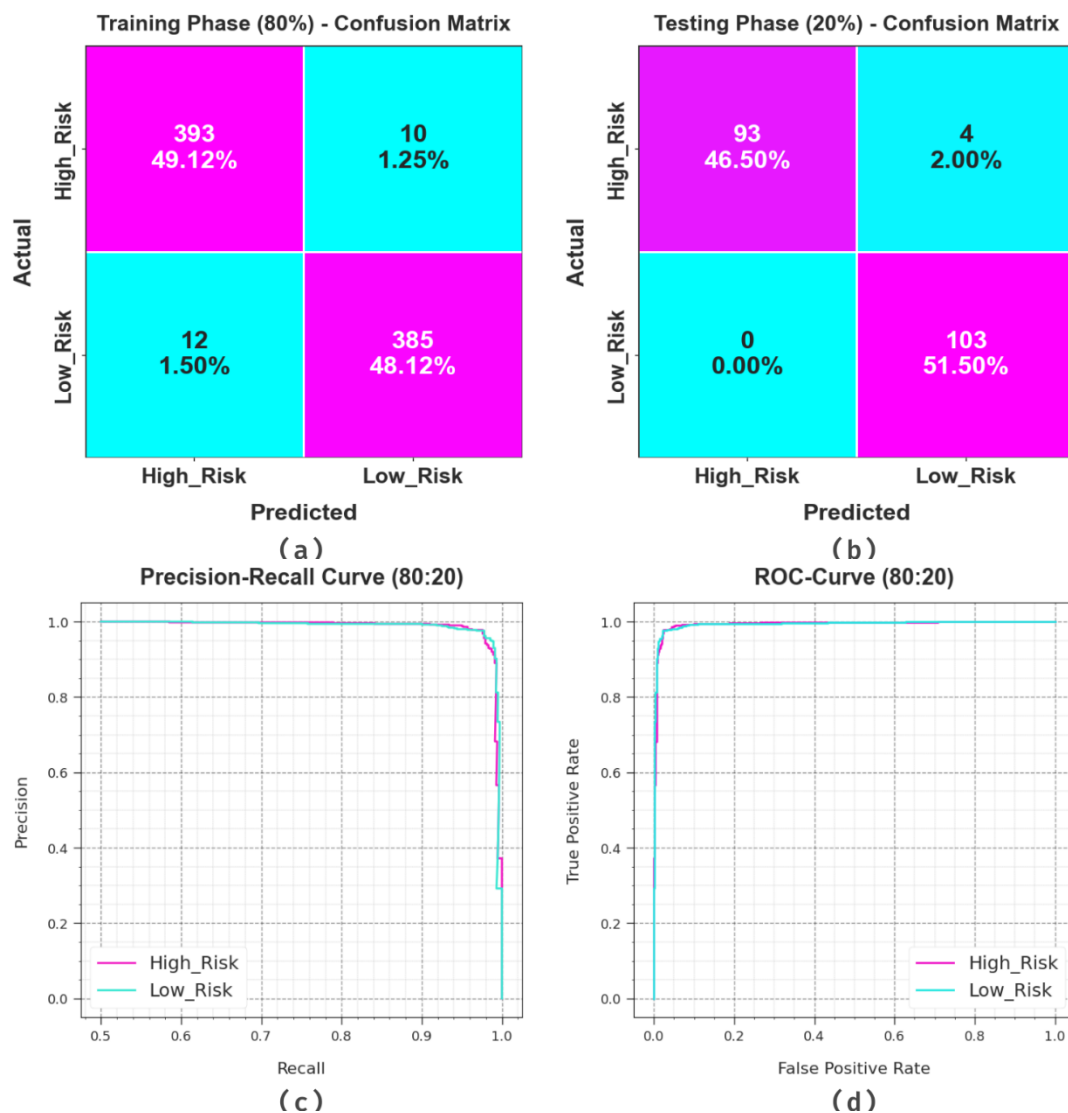


Fig. 5. 80:20 of TRPH/TSPH of (a-b) Confusion matrices (c) PR-curve and (d) ROC-curve

In Table 2 and Fig. 6, the SD detection analysis achieved by the AOADL-SDDC method on 80:20 of TRPH/TSPH can be confirmed. The obtained values showcase the increased performance of the AOADL-SDDC system under the SD detection process. According to 80% of TRPH, the AOADL-SDDC algorithm offers average $accu_y$, $prec_n$, $reca_l$, F_{score} , and MCC of 97.25%, 97.25%, 97.25%, 97.25%, and 94.50%. In the meantime, with 20% of TSPH, the AOADL-SDDC technique gives average $accu_y$, $prec_n$, $reca_l$, F_{score} , and MCC of 98.00%, 98.13%, 97.94%, 97.99%, and 96.07%, correspondingly.

Table 2 SD detection outcomes of AOADL-SDDC model on 70:30 of TRPH/TSPH

Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}	MCC
TRPH (80%)					
High_Risk	97.25	97.04	97.52	97.28	94.50
Low_Risk	97.25	97.47	96.98	97.22	94.50
Average	97.25	97.25	97.25	97.25	94.50
TSPH (20%)					
High_Risk	98.00	100.00	95.88	97.89	96.07
Low_Risk	98.00	96.26	100.00	98.10	96.07
Average	98.00	98.13	97.94	97.99	96.07

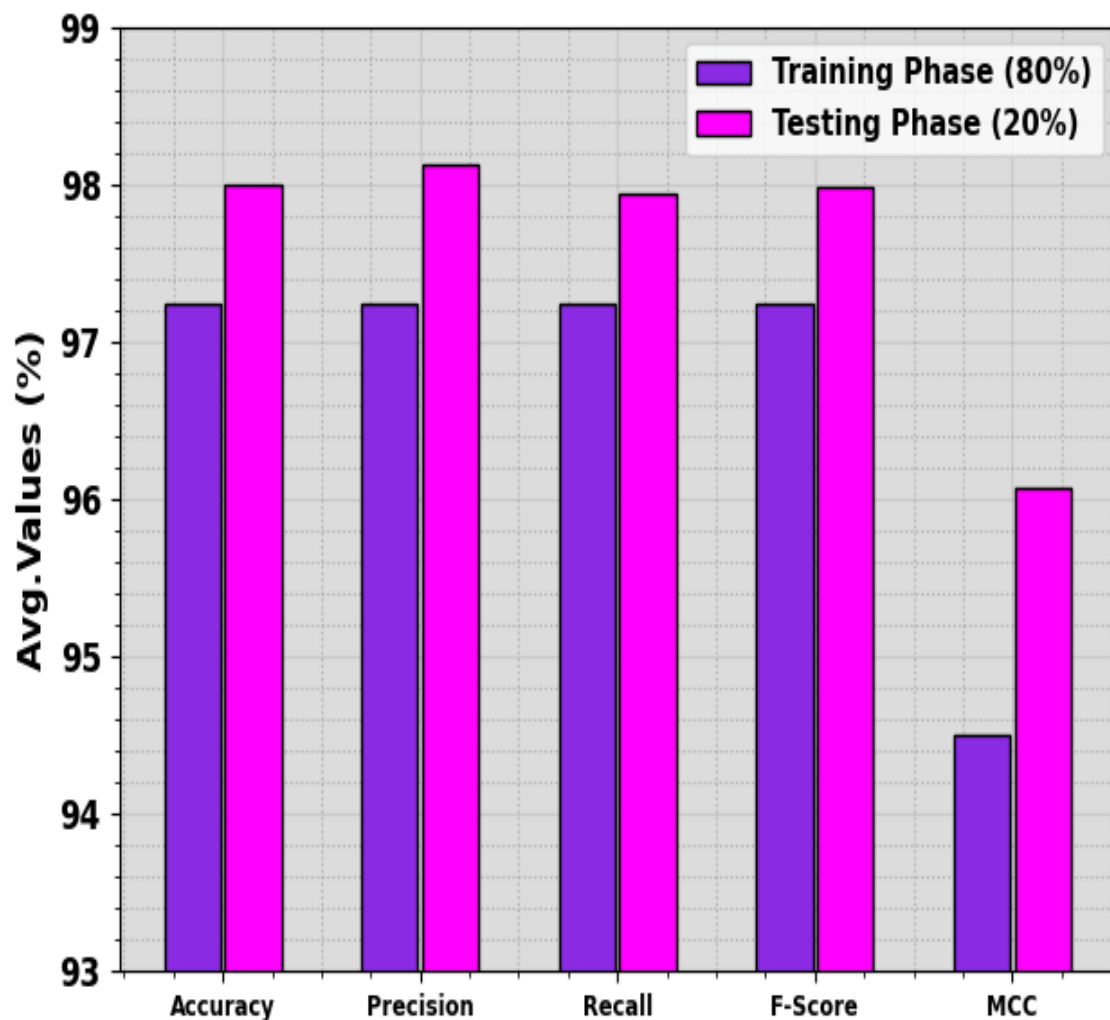


Fig. 6. Average outcomes of AOADL-SDDC algorithm on 80:20 of TRPH/TSPH

Fig. 7 demonstrates the classifier analysis of the AOADL-SDDC system under 70:30 of TRPH/TSPH. Figs. 7a-7b represents the confusion matrix created by the AOADL-SDDC technique. This figure shows that the AOADL-SDDC method can be appropriately identified and categorized with high_risk and low_risk classes. Then, Fig. 7c

displays the PR analysis of the AOADL-SDDC method. This figure specified the AOADL-SDDC methodology achieves boosted PR performance with every class. In the same way, Fig. 7d illustrates the ROC analysis of the AOADL-SDDC technique. The figure displayed that the AOADL-SDDC system gives proficient outcomes with greater ROC values with diverse class labels.

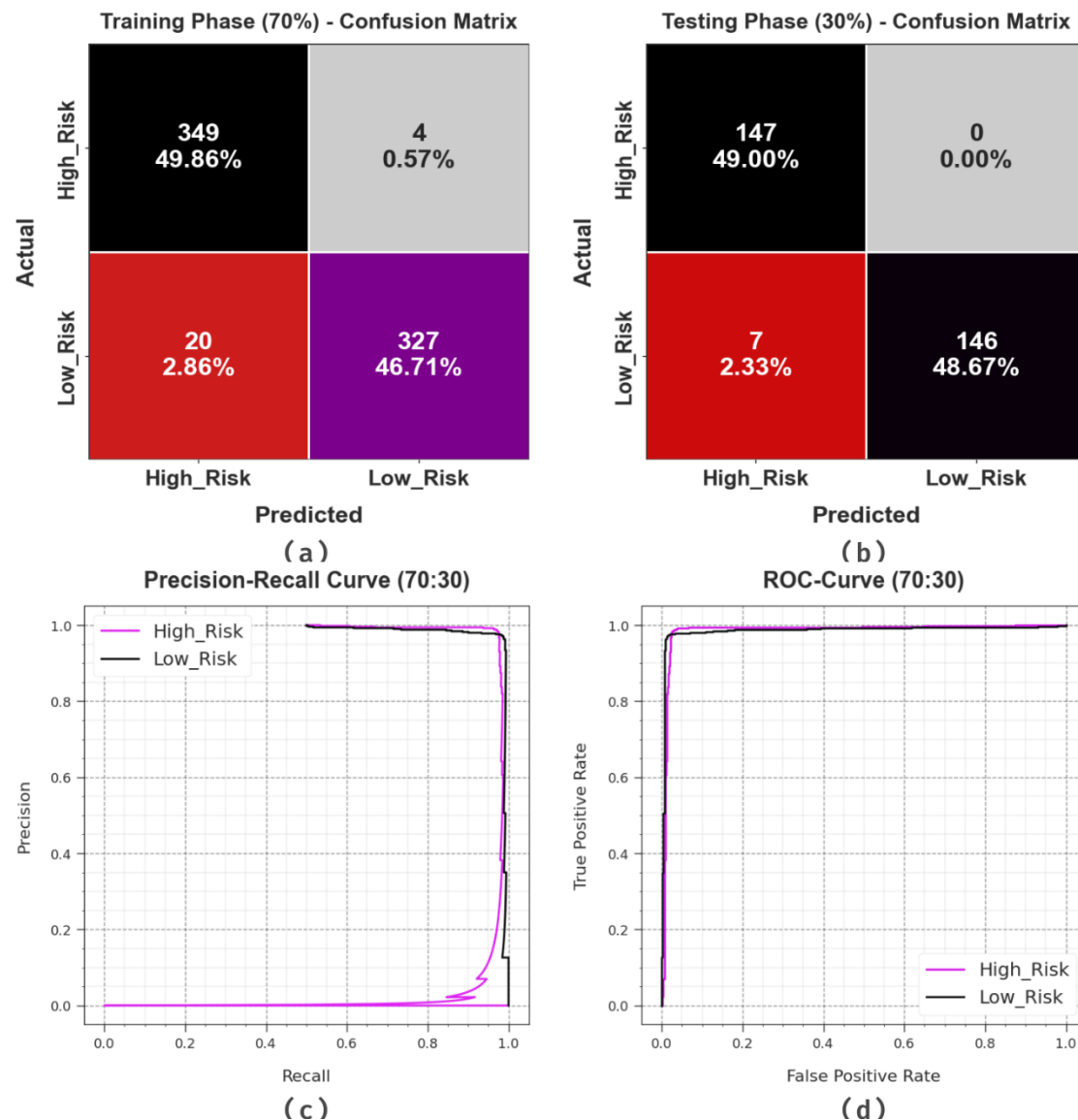


Fig. 7. 70:30 of TRPH/TSPH of (a-b) Confusion matrices (c) PR-curve and (d) ROC-curve

In Table 3 and Fig. 8, the SD detection analysis accomplished by the AOADL-SDDC technique on 70:30 of TRPH/TSPH can be examined. The obtained result displays the increased performance of the AOADL-SDDC system on the SD detection process. Based on 70% of TRPH, the AOADL-SDDC algorithm provides average $accu_y$, $prec_n$, $reca_l$, F_{score} , and MCC of 96.55%, 96.69%, 96.55%, 96.57%, and 93.24%. Besides, 30% of TSPH, the AOADL-SDDC technique offers average $accu_y$, $prec_n$, $reca_l$, F_{score} , and MCC of 97.71%, 97.73%, 97.71%, 97.67%, and 95.44%, respectively.

Table 3 SD detection analysis of AOADL-SDDC model under 70:30 of TRPH/TSPH

Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}	MCC
TRPH (70%)					

High_Risk	98.87	94.58	98.87	96.68	93.24
Low_Risk	94.24	98.79	94.24	96.46	93.24
Average	96.55	96.69	96.55	96.57	93.24
TSPH (30%)					
High_Risk	100.00	95.45	100.00	97.67	95.44
Low_Risk	95.42	100.00	95.42	97.66	95.44
Average	97.71	97.73	97.71	97.67	95.44

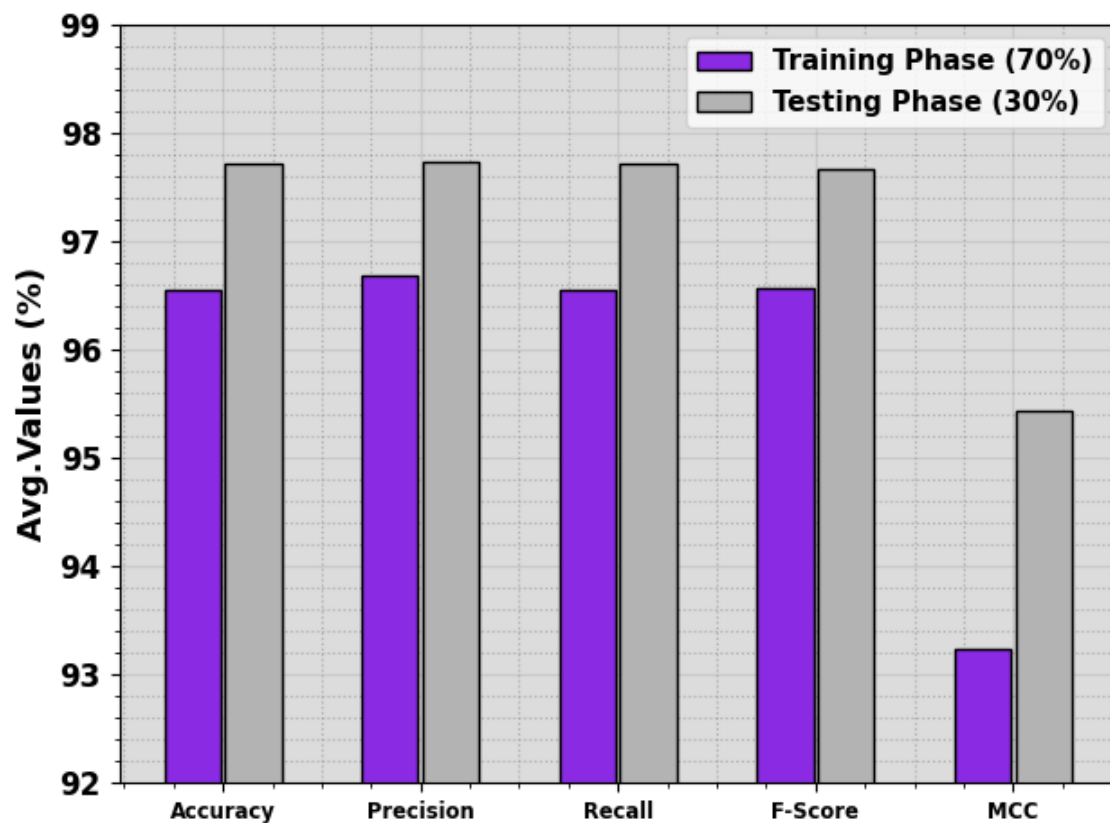
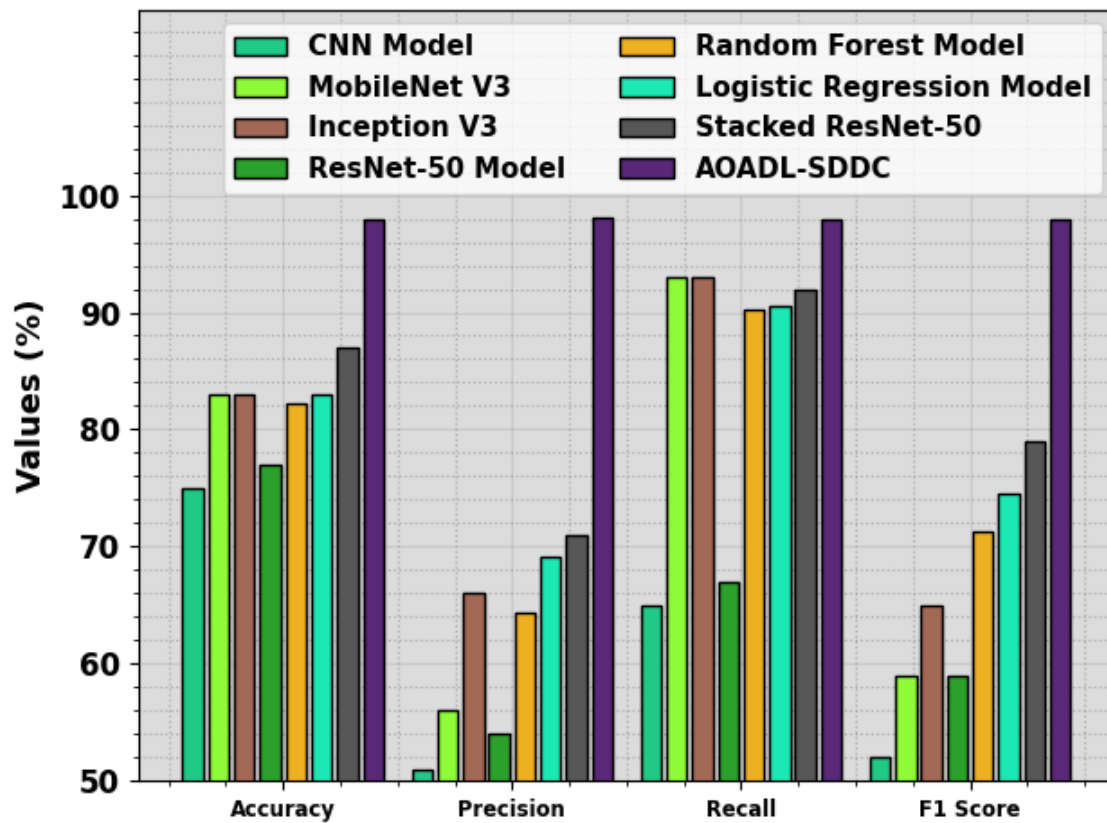


Fig. 8. Average outcomes of AOADL-SDDC system at 70:30 of TRPH/TSPH

For more validation of the improvement of the AOADL-SDDC methodology, an extensive comparison analysis can be described in Table 4 and Fig. 9 [12]. The accomplished result revealed that the AOADL-SDDC model displays excellent performance over other systems. According to $accu_y$, the AOADL-SDDC system gives boosting $accu_y$ of 98% but, the CNN, MobileNet, Inception v3, ResNet-50, RF, LR, and Stacked ResNet-50 algorithms lead to decreased $accu_y$ of 75%, 83%, 83%, 77%, 82.30%, 82.98%, and 87%, respectively. Similarly, on $prec_n$ the AOADL-SDDC model provides raised $prec_n$ of 98.13% whereas the CNN, MobileNet, Inception v3, ResNet-50, RF, LR, and Stacked ResNet-50 approaches obtains minimized $prec_n$ of 51%, 56%, 66%, 54%, 64.40%, 69.12%, and 71%. Then, on $reca_l$ the AOADL-SDDC method gets higher $reca_l$ of 97.94% however, the CNN, MobileNet, Inception v3, ResNet-50, RF, LR, and Stacked ResNet-50 techniques acquire lesser $reca_l$ of 65%, 93%, 93%, 67%, 90.27%, 90.54%, and 92%, respectively.

Table 4 Comparison analysis of the AOADL-SDDC model with other approaches

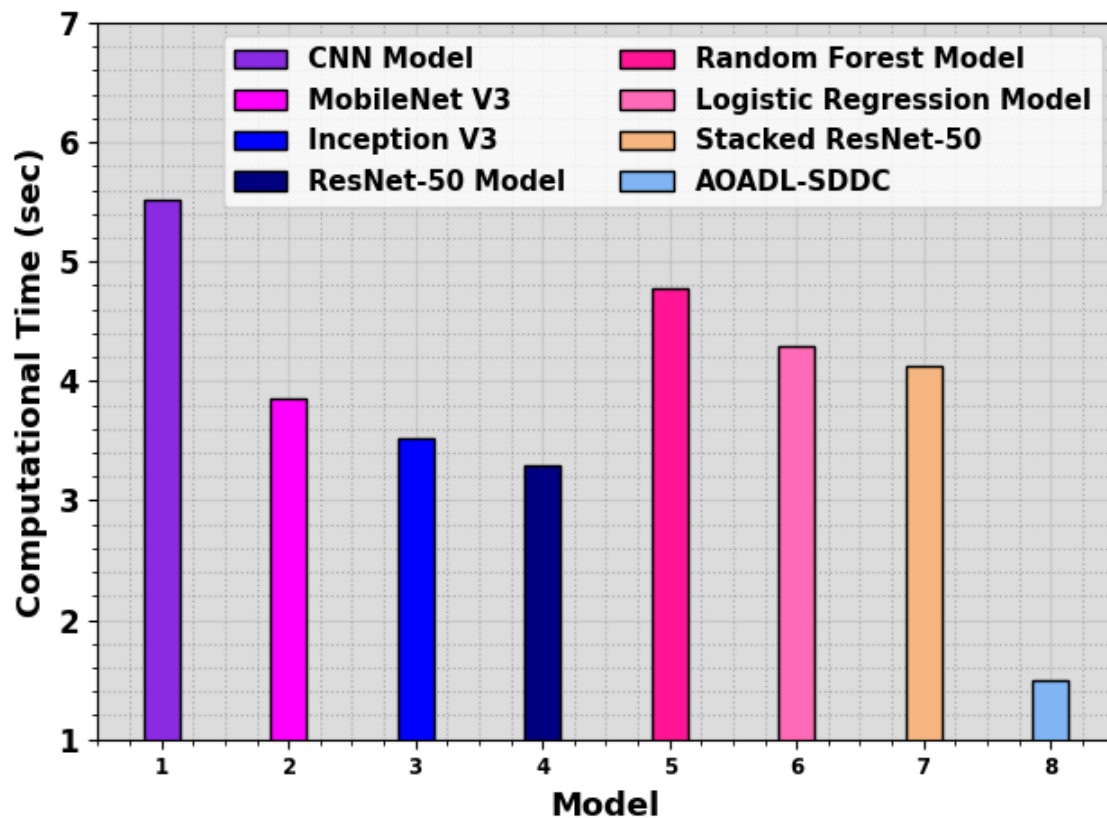
Model	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}
CNN Model	75.00	51.00	65.00	52.00
MobileNet V3	83.00	56.00	93.00	59.00
Inception V3	83.00	66.00	93.00	65.00
ResNet-50 Model	77.00	54.00	67.00	59.00
Random Forest Model	82.30	64.40	90.27	71.23
Logistic Regression Model	82.98	69.12	90.54	74.57
Stacked ResNet-50	87.00	71.00	92.00	79.00
AOADL-SDDC	98.00	98.13	97.94	97.99

**Fig. 9.** Comparison analysis of the AOADL-SDDC model with other algorithms

Finally, a wide-ranging comparative computation time (CT) analysis of the AOADL-SDDC technique can be presented in Table 5 and Fig. 10. The acquired outcomes display the higher performance of the AOADL-SDDC methodology with a decreased CT of 1.50s.

Table 5 CT analysis of the AOADL-SDDC model with other existing systems

Model	Computational Time (sec)
CNN Model	5.52
MobileNet V3	3.86
Inception V3	3.52
ResNet-50 Model	3.30
Random Forest Model	4.77
Logistic Regression Model	4.29
Stacked ResNet-50	4.13
AOADL-SDDC	1.50

**Fig. 10.** CT outcome of the AOADL-SDDC algorithm with other models

Alternatively, the CNN, MobileNet, Inception v3, ResNet-50, RF, LR, and Stacked ResNet-50 systems gains improved CT values of 5.52s, 3.86s, 3.52s, 3.30s, 4.77s, 4.29s, and 4.13s. These accomplished outcomes confirmed the superior detection performance of the AOADL-SDDC method with respect to various metrics.

5. Conclusion

In this study, we develop a new AOADL-SDDC method. The purpose of the AOADL-SDDC technique is to exploit hyperparameter tuned DL model for SD recognition. The AOADL-SDDC technique comprises three major processes namely pre-processing, segmentation, and classification. At the pre-processing stage, the quality of the input images can be enhanced by the use of WF based noise removal and DHE based contrast improvement.

Besides, the segmentation process takes place where the UNet++ with RMSProp optimizer and Manhattan distance based pedestrian distance calculation are involved. Finally, the AOA with BiConvLSTM model can be exploited for accurate detection and classification of the SD. A detailed set of experiments was carried out to examine the performance of the AOADL-SDDC system. The stimulation values stated that the AOADL-SDDC technique reaches superior performance over other models.

References

1. Yang, D., Yurtsever, E., Renganathan, V., et al.: A vision-based social distancing and critical density detection system for COVID-19. *Sensors* 21(4608), 1–15 (2021)
2. Punns NS, Sonbhadra SK, Agarwal S. Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques. *arXiv preprint arXiv:200501385*. 2020.
3. Ainslie KE, Walters CE, Fu H, Bhatia S, Wang H, Xi X, et al. Evidence of initial success for China exiting COVID-19 social distancing policy after achieving containment. *Wellcome Open Research*. 2020; 5.
4. Rezaei, M., Azarmi, M.: DeepSOCIAL: social distancing monitoring and infection risk assessment in COVID-19 pandemic. *Appl. Sci.* 10(7514), 1–29 (2020)
5. Adolph C, Amano K, Bang-Jensen B, Fullman N, Wilkerson J. Pandemic politics: Timing state-level social distancing responses to COVID-19. *medRxiv*. 2020.
6. Vinitha, V., Velantina, V.: Social distancing detection system with artificial intelligence using computer vision and deep learning. *Int. Res. J. Eng. Technol. (IRJET)* 7(8), 4049–4053 (2020)
7. Prem K, Liu Y, Russell TW, Kucharski AJ, Eggo RM, Davies N, et al. The effect of control strategies to reduce social mixing on outcomes of the COVID-19 epidemic in Wuhan, China: a modelling study. *The Lancet Public Health*. 2020
8. Saponara, S., Elhanashi, A., Gagliardi, A.: Implementing a real-time, AI based, people detection and social distancing measuring system for Covid-19. *J. Real-Time Image Process.* 18(6), 1937–1947 (2021)
9. Arya, S., Patil, L., Wadekanokar, A., et al.: Study of various measure to monitor social distancing using computer vision: a review. *Int. J. Eng. Res. Technol. (IJERT)* 10(5), 329–326 (2021)
10. Cobb J, Seale M. Examining the effect of social distancing on the compound growth rate of SARS-CoV2 at the county level (United States) using statistical analyses and a random forest machine learning model. *Public Health*. 2020.
11. Dave, B., Verma, J.P., Jain, R. and Nayyar, A., 2023. Multi-object Detection: A Social Distancing Monitoring System. In *Object Tracking Technology: Trends, Challenges and Applications* (pp. 221-248). Singapore: Springer Nature Singapore.
12. Walia, I.S., Kumar, D., Sharma, K., Hemanth, J.D. and Popescu, D.E., 2021. An integrated approach for monitoring social distancing and face mask detection using stacked Resnet-50 and YOLOv5. *Electronics*, 10(23), p.2996.
13. Hermina, J., Karpagam, N.S., Deepika, P., Jeslet, D.S. and Komarasamy, D., 2022. A Novel Approach to Detect Social Distancing Among People in College Campus. *International Journal of Intelligent Systems and Applications in Engineering*, 10(2), pp.153-158.
14. Oztel, I., Yolcu Oztel, G. and Akgun, D., 2023. A hybrid LBP-DCNN based feature extraction method in YOLO: An application for masked face and social distance detection. *Multimedia Tools and Applications*, 82(1), pp.1565-1583.
15. Shete, I., 2020. *Social distancing and face mask detection using deep learning and computer vision* (Doctoral dissertation, Dublin, National College of Ireland).
16. Meivel, S., Sindhwani, N., Anand, R., Pandey, D., Alnuaim, A.A., Altheneyan, A.S., Jabarulla, M.Y. and Lelisho, M.E., 2022. Mask detection and social distance identification using internet of things and faster R-CNN algorithm. *Computational Intelligence and Neuroscience*, 2022.
17. Kapoor, N., Bär, A., Varghese, S., Schneider, J.D., Hüger, F., Schlicht, P. and Fingscheidt, T., 2021, July. From a fourier-domain perspective on adversarial examples to a wiener filter defense for semantic segmentation. In *2021 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.

-
18. Saravanan, S. and Karthigaivel, R., 2021. A fuzzy and spline based dynamic histogram equalization for contrast enhancement of brain images. *International Journal of Imaging Systems and Technology*, 31(2), pp.802-827.
 19. Lu, Y., Qin, X., Fan, H., Lai, T. and Li, Z., 2021. WBC-Net: A white blood cell segmentation network based on UNet++ and ResNet. *Applied Soft Computing*, 101, p.107006.
 20. Arias, F.X., Sierra, H., Jimenez-Rodriguez, L.O. and Arzuaga, E., 2017. Supervised sparse-representation classification on hyperspectral images using the city-block distance to improve performance.
 21. Wang, Y., Wang, W., Zang, H. and Xu, D., 2023. Is the LSTM Model Better than RNN for Flood Forecasting Tasks? A Case Study of HuaYuankou Station and LouDe Station in the Lower Yellow River Basin. *Water*, 15(22), p.3928.
 22. Chang, Y. and Luo, B., 2019. Bidirectional convolutional LSTM neural network for remote sensing image super-resolution. *Remote Sensing*, 11(20), p.2333.
 23. Hashim, F.A., Khurma, R.A., Albashish, D., Amin, M. and Hussien, A.G., 2023. Novel hybrid of AOA-BSA with double adaptive and random spare for global optimization and engineering problems. *Alexandria Engineering Journal*, 73, pp.543-577.
 24. <https://github.com/ChargedMonk/Social-Distancing-using-YOLOv5>