Journal of Intelligent Systems and Internet of Things
Vol. 16, No. 01, PP. 28-40, 2025
DOI: https://doi.org/10.54216/JISIoT.160103

Leveraging Marine Predators Algorithm with Deep Learning Object Detection for Accurate and Efficient Detection of Pedestrians

Hima Bindu Gogineni[1,*], Hemanta Kumar Bhuyan[2], E. Laxmi Lydia[3,4]

[1]PhD Scholar, Department of Information Technology, Vignan's Foundation for Science, Technology & Research University, Guntur, Andhra Pradesh, India

[2]Associate Professor, Department of Information Technology, Vignan's Foundation for Science, Technology & Research University, Guntur, Andhra Pradesh, India

[3]Department of Information Technology, VR Siddhartha Engineering College (A), Siddhartha Academy of Higher Education (Deemed to be University), Vijayawada, India

[4]Department of Computer Science and Engineering, Vignan's Institute of Information Technology (Autonomous), Vishakhapatnam, India

Emails: goginenibindu9@gmail.com; hmb.bhuyan@gmail.com; elaxmi2002@yahoo.com

Abstract

Pedestrian detection using object detection and deep learning has been found to be effective method for identifying pedestrians in video frames or images accurately. It is more commonly used in many real-time applications, such as security observing systems, autonomous driving systems, and robotics. The combination of deep learning techniques and object detection algorithms allows efficient and robust detection of pedestrians in several real-time scenarios. However, it is necessary to improve the detection efficacy for complex environments such as cases with worse visibility due to weather or daytime, crowd scenes, and rare pose samples. Continuous improvement and research in DL algorithms, dataset collection, and TRA models contribute to accelerating the robustness and acc of pedestrian detection systems. Therefore, this research models a novel marine predator algorithm with DL-based pedestrian detection and classification (MPADLB-PDC) method. The objective of the MPADLB-PDC system lies in the accurate recognition and identification of pedestrians. To achieve this, the MPADLB-PDC technique involves two major processes, namely object detection and classification. In the first stage, the MPADLB-PDC technique uses an improved YOLOv7 object detector for the recognition of the objects in the frame. Next, in the second stage, the ensemble classifier comprises three classifiers such as deep feed-forward neural networks (DFFNNs), extreme learning machine (ELM), and long short-term memory (LSTM). To improve the recognition performance of the ensemble classifier, the MPA is used to optimally select the parameters related to it. The simulation outcome of the MPADLB-PDC technique was authorized on the pedestrian database, and the outcome can be studied in terms of various aspects. The experimentation values validated the better outcome of the MPADLB-PDC approach compared to other approaches.

Received: November 19, 2024 Revised: January 01, 2025 Accepted: January 30, 2025

Keywords: Computer vision; Deep learning; Pedestrian detection; Marine predator's algorithm; Ensemble learning

1. Introduction

At present, intelligent transportation methods was developed to decrease the capacity of traffic in metropolitan cities, decrease the rate of injuries and accidents, and deaths, minimize fuel consumption, decrease environmental pollution, etc [1]. Such systems used diverse technologies (including deep learning (DL), IoT, data mining, image processing, neural networks (NN), and machine learning (ML) for different applications [2]. Conversely, large automotive and technology companies (e.g. Tesla and Google) are trying to mature self-driving smart cars that can

offer safe travel for individuals if the driver is tired and can save the life of commuters and drivers; with automatic control, car accidents can be prevented [3]. Such e vehicles should be equipped with sensors to sense the environment, find objects near the car, update the driver, and have actuators to achieve real-time methods if the driver is drowsy or the driver fails to pay attention to threats to evade accidents [4].

Pedestrian recognition approaches are classified as 2 categories deep features-based and hand-crafted features based. In handcrafted features based, the handcrafted features, namely Integral Channel Features (ICF) and Histogram of Oriented Gradients (HOG), are mined to train classifier [5]. Such approaches are adequate for simple cases. However, the performance is not satisfactory, and the efficacy is low [6]. With the fast growth of DL, particularly the proposal of generic object recognition, DL-related techniques for pedestrian recognition attain substantial enhancements in the context of acc and speed [7]. However, existing pedestrian detection performance is not compared with human perception. Current deep convolutional neural networks (DCNNs)-relevant approaches are leading to important progress in pedestrian detection. It remains a problem as pedestrians frequently occlude towards each other. Over a couple of years, with advances, pedestrian recognition was applied effectually in particular practical application systems in a few cases [8]. The success of this process is recognized in 2 phases: advances in the DCNN that had a direct impact on numerous CV tasks that, includes pedestrian recognition and database gathered efforts carried out by many authors. In addition, enrichments in detection methods are followed by the publication of datasets that are more complicated [9]. Concerning pedestrian detection, openly accessible benchmark data such as KITTI, CityPersons, and Caltech-USA have fortified growth in and interest in pedestrian recognition researches [10].

This paper presents an efficient marine predator algorithm with deep learning-based pedestrian detection and classification (MPADLB-PDC) technique. The MPADLB-PDC technique involves two major processes, namely object detection and classification. In the first stage, the MPADLB-PDC technique uses improved YOLOv7 object detector for the recognition of the objects from the frame. Besides, ensemble classifiers comprise three classifiers such as deep feed-forward neural networks (DFFNNs), extreme learning machine (ELM), and long short-term memory (LSTM). Lastly, the MPA was used to optimum select the parameters compared to the ensemble models. The experimental outcome of the MPADLB-PDC approach was validated on pedestrian dataset, and the outcome can be studied interms of various aspects.

2. Literature Review

Li et al. [11] proposed an enhanced multiscale YOLOv5s algorithm, and CARAFE element (a content-aware reassembly of feature) was presented in the feature fusion portion to boost the feature fusion. Rather than the original convolution infrastructure, an SPD-Conv CNN Element was devised to enrich the computational efficiency. Lastly, introduced a NAM (normalization-based attention element) for focusing on user data in TRA and enhancing detection acc. In [12], the authors devised a more robust method of attribute recognition and pedestrian tracking, easing the study of pedestrian walking behaviour. Certain limitations of contemporary schemes can be concluded based on several improvement approaches modelled: 1) a probation system for strong uniqueness matching, 2) including higher-level pedestrian features to boost pedestrian chasing, and 3) a similarity measure compiling multiple cues for uniqueness matching. Ansarnia et al. [13] concentrate on examining video streams taken in vertically fixed camera and making relative road user recognition.

Kolluri and Das [14] developed an IPDC-HMODL method (intellectual multimodal pedestrian recognition and identification utilizing hybrid metaheuristic optimizer with DL). This has followed 3 processes. Initially, the presented method makes use of multimodal object detectors through Retina-Net and YOLOv5 approach. Then, the presented model implements KELM method abbreviated as kernel ELM, for classifying pedestrians. Eventually, for optimum parameter adjustment, hybrid salp swarm optimizer (HSSO) method was employed. Sharma et al. [15] define many DL methods and their features for object recognition in videos and pictures. Other object detection applications, namely autonomous driving, face identification, and pedestrian detection, are explored. Object recognition refers to a set of tasks that identify items in digital images. Region-based CNN is a class of methods to handle object recognition and localization problems that can be optimized for model performance. In [16], designed DL-related Multi-Object Tracking method AerialMPTNet compiles appearance, graphical, and temporal data by means of a Graph CNN, SNN, and LSTM for stable tracking.

Fayyaz and Johnson [17] introduced a method that incorporates DL technology with the current Vision mechanism. The presented sensing system must localize and identify specific objects. Various approaches for TRA a DL method are deliberated with corresponding outcomes. Zhou et al. [18] devised an XDRNet method, which is a DL-based pedestrian and vehicle Indoor positioning approach. Then, pedestrian behaviour and driving behaviour are differentiated by DNN approach. In this study, to make the method a superior one on smartphones, a lightweight network structure was utilized.

3. The Proposed Pedestrian Detection Model

In this manuscript, we have provided an automated pedestrian recognition model called the MPADLB-PDC technique, which mainly focuses on the accurate recognition and classification of pedestrians. Primarily, the MPADLB-PDC technique involves two major processes, namely improved YOLOv7-based object recognition and ensemble classification. Fig. 1 illustrates the workflow of the MPADLB-PDC system.

A. Object Detection: Improved YOLO-v7 Model

Initially, the MPADLB-PDC approach uses improved YOLOv7 object detector for the recognition of the objects in the frame. The YOLOv7 model primarily consists of five different components: Neck, Head, input layer, loss function, and backbone network [19]. In the input layer, we adopted adaptive image scaling, data augmentation, and adaptive anchor box calculation models. The original YOLOv7 backbone network has overall of fifty elements, involving the MP1 element, CBS element, and ELAN element. The original CBS element includes a SiLU, a standard convolution, and a BN layer. Nevertheless, typical convolution takes a lot of computing resources. The initial stage is to exchange the original CBS elements with cheapest Ghost convolution elements to accomplish lightweight deployment. Next, the original ELAN element encompasses six CBS elements; thus, it exchanges the CBS element with the better GhostCBS element and increases the CBAM attention elements to generate novel CG-ELAN elements. It reduces the model inference time through the shared attention element and efficiently extracts the local infrastructure of images with shift convolution element. Eventually, in order to make the original image, a CBAM model was proposed. The channel and spatial attention elements are taken into account for enhancing the efficiency of long-range self-attention of networks but preserving the parameter count. The MP element includes a CBS and a Maxpool that can be split into $MP1$ and $MP2$. The main difference between them is that the channel of previous one remains unchanged, whereas the channel of last one is doubled. A similar improvement was applied for replacing the CBS elements with Ghost-CBS elements to optimize the sampling network effects.

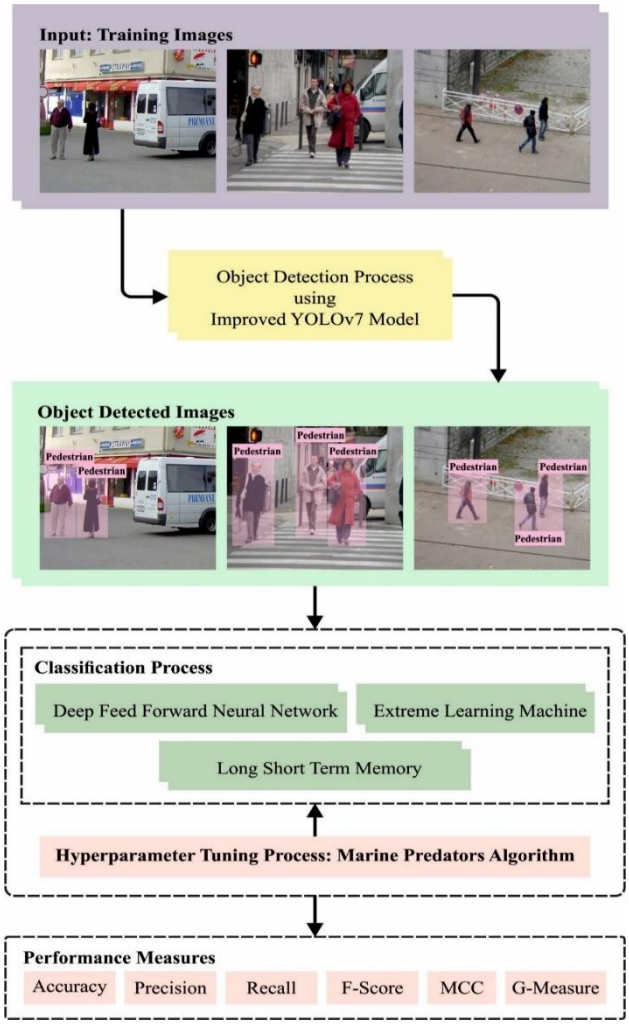

Figure 1. Workflow of MPADLB-PDC approach

Combine the advantages of the Head part of YOLOv7 with PAN and the FPN to procedure PA-FPN architecture, making the feature map of dissimilar levels to accomplish the effects of effective fusion. The initial progress made is to exchange the CBS element with the Ghost-CBS element. Next, add a smaller object detection layer that is used for generating 160x160 feature maps and extending the recognition range of the network. Even though these models will bring a slight improvement in the count of detection and computation speed, they still efficiently enhance the recognition performance. Four RepConv elements are applied at the end of the Head, which could not be directly linked to re-parameterize the network. Here, the re-parameterization represents the usage of multi-branch structure during TRA that is corresponding to multi-scale NNs contributing to learning together, thus enhancing the recognition performance of the model. Similar to YOLOv5, YOLOv7 implements a loss calculation method in terms of the loss function that can be split into coordinate loss, object confidence loss, and classification loss. The co-ordinate loss exploits the existing outstanding CIoU loss that considers the centre distance, overlapping area, aspect ratio, and other factors, and could enhance the recognition performance in resolving the problems of non-overlapping detection box.

B. Pedestrian Classification: Ensemble Process

In this study, ensemble classifier is utilized for the identification and classification of pedestrians. Here, weighted majority voting (WMV) model is used for ensemble process. A certain classifier is better qualified than others are. Weighting the decision of the classifier through the prediction acc might strengthen the decision of qualified classifier, making it possible to provide additional importance to the decision during the vote and subsequently enhance the overall performance. Here, every vote is weighted by the acc value of the classifier prediction, which can be represented as Acc. The count of overall votes for the class $c_k$in (1) is rewritten as follows:

$$T_k = \sum_{l=1}^{M} Acc(A_l) \times F_k(c_l) \tag{1}$$

The class receiving the greatest total weight is chosen.

1. DFFNN Model

DFFNN is the more commonly used ML algorithm in predicting volatility. It is single input layer with m predictors, x_i, 4 hidden layers (HLs), each comprising n neurons from which all the neurons can perform a nonlinear conversion of outputs in prior layer, and lastly, single resultant layer which generates the value of y equivalent to the predictive value of prior volatility [20]. Note that many architectures of these models interms of count of neurons in all the layers, count of HL, and so on, are used to search for the better architecture for all the cryptocurrency's data:

$$y = f_l\left(\ldots f_1\left(\sum_{\mathrm{j}=1}^{m_1} w_{jk}^{(1)} . f_0\left(\sum_{i=1}^{m_0} w_{i_j}^{(0)} . x_i + b_j^{(0)}\right) + b_k^{(1)}\right)\ldots\right) \tag{2}$$

Where $b_j^{(l)}$ shows the biased term for j^{th} neurons in layer l^{th}, $f_l(\cdot)$ shows the activation function from layer l^{th} and also controls the magnitude of output from all the neurons. w_{ij} refers to the weight or parameter which links j^{th} neuron to i^{th} neuron from the prior layer. m_l shows the neuron counts from the layer l^{th}. The procedure of TRA the DFFNN is that the optimizer technique (back-propagation) alters the unknown biases and weights via iteration process such that the value of lost functions is minimalized. Fig. 2 showcases the architecture of DFFNN.

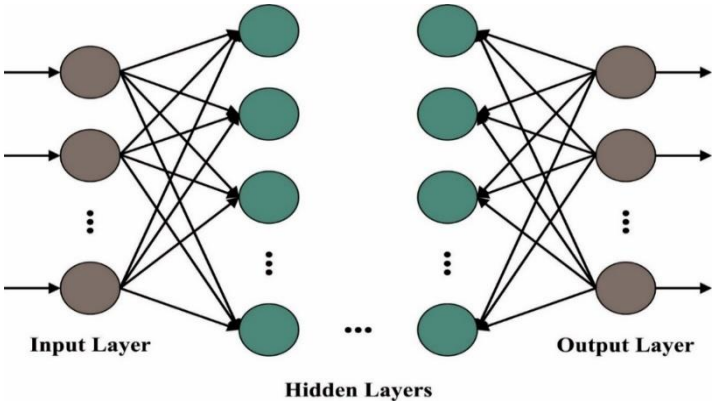

Figure 2. DFFNN structure

2. ELM Model

ELM is basically an FFNN in which the thresholds and input weights are initialized randomly to enhance the TRA efficacy for use in many cases [21]. Assume N samples (X_i, X_j) and the input layer was mapped by the active function expressed using Eq. (3):

$$\sum_{i=1}^{L} \beta_i f\left(\omega_i x_j + b_i\right) = o_j, \tag{3}$$

Where b_i refers to the bias of the hidden node, L denotes the amount of hidden layer (HL) nodes, x_1, x_2, ..., x_i, ω_i shows the input weight, and β_i indicates the output weight.

Later, the network TRA realizes learning with zero errors,

$$\sum_{j=1}^{N} \| o_j - t_j \| = 0, \tag{4}$$

and there are w_i, β_i, and b_i, making the succeeding condition hold:

$$\sum_{i=1}^{N} \beta_i f\left(\omega_i X_j + b_j\right) = t_j, \tag{5}$$

In Eq. (5), b_i denotes the threshold between implicit layers, β_i shows the output weight, and t_j indicates the output value:

$$H\beta = T,$$
$$H = \begin{bmatrix} f(\omega_1 X_1 + b_1) & \dots & f(\omega_N X_1 + b_L) \\ \vdots & \ddots & \vdots \\ f(\omega_1 X_N + b_1) & \dots & f(\omega_L X_N + b_L) \end{bmatrix}. \tag{6}$$

Consider the ELM principle; the thresholds and weights of input are assigned arbitrarily, where the weights among the output layer as well as HL are attained by the solution to the system of equations:

$$\beta^* = H^*.T, \tag{7}$$

Eq. (7) $*$ indicates the Moore-Penrose pseudo-inverse of the corresponding matrix.

3. LSTM Model

LSTM is a kind of RNN where task displays long-term dependency if predictive of required output at t time relies on input provided at prior time T [22]. Gradient-based learning algorithm, including RNN, faces difficulty in implementing this task, and the parameter settles in a sub-optimum solution that only considers a short-term dependency, not a long-term dependency. The dissimilar parts of the LSTM unit consist of 3 gates such as forget, input, and output gates hidden state (h) and cell state (C). This gate controls the data flow in LSTM.

Forget gate: first, LSTM adopts if the data from prior timestamp must be retained or not.

$$f_t = \sigma\left(X_t.U_f + H_{t-1}.W_f\right) \tag{8}$$

In Eq. (8), X_t denotes input to the present timestamp, t, U_f indicates weight related to the input, H_{t-1} shows the HL of the $t-1$ prior timestamp, and W_f represents the weight matrix related to HL. A sigmoid function was used to f_t, and value developed 1. A value of 1 denotes it will keep all the data from $t-1$, and value of 0 implies that the network forgets everything.

Input gate: this gate quantifies the significance of novel data that are entered into unit.

$$i_t = \sigma\left(X_t.U_j + H_{t-1}.W_i\right) \tag{9}$$

Where U_i and W_i denote the weight matrix related to input and HLs, another sigmoid function is used for it, and its value is between zero and one. Novel data is subtracted or added to the cell state of the LSTM. The tanh function can be utilized for making the input data among -1 for subtraction of data in cell state and 1 to add data

to the cell state; however, the novel data, N_t, was updated using forget and input gates before entering the cell state.

$$N_t = tanh(X_t.U_c + H_{t-1}.W_c) \quad (10)$$

$$C_t = f_t.C_{t-1} + i_t.N_t \quad (11)$$

Where C_{t-1} represents the cell state at the $t-1$ prior timestamp.

Output gate: this gate determines that data should be output as HL at t time.

$$O_t = \sigma(X_t.U_o + H_{t-1}.W_o) \quad (12)$$

$$h_t = O_t.\tanh(\,) \quad (13)$$

4. MPA-based Parameter Optimization

To adjust the parameters related to the ensemble classifier models, the MPA is used. Like other metaheuristics, MPA splits the optimization algorithm into exploitation and exploration [23]. The optimization algorithm starts by creating a primary population of searching agents at random. Afterwards, a Fitness function is evaluated, and the better carrying out performance acts as a marine predator, whereas the remaining solution serves as prey. Here, the searching space has single MP (the better solution) and multiple prey. Based on velocity ratios (three encounter policies) discussed previously are modeled in MPA by spilling the optimizer algorithm as 3 equivalent stages. During the first third, prey (solution candidate) updates the position with regard to MP (optimum result) from Brownian fashion. During the second phase, the solution candidate was divided in half, and every half had its location vector upgraded in Brownian or Lévy manner with regard to top solutions. Lastly, During the third iteration, the MP has its location vector multiplied with Levy operators, and the prey updates the solution. The better solution is updated continuously for all the iteration cycles.

The algorithm starts with the uniform distribution population of the searching agent over the computation field using Eq. (14).

$$\vec{X} = \overrightarrow{X_L} + \overrightarrow{rand} \otimes \left(\overrightarrow{X_U} - \overrightarrow{X_L}\right)\ i = 1, \dots, PopSize \quad (14)$$

Where $\overrightarrow{X_L}$ and $\overrightarrow{X_U}$ represent the lower as well as upper boundaries, correspondingly, The better performance from the population was represented as $\overrightarrow{Elite}$, and each solution candidate updates the position. $\overrightarrow{rand}$ indicates the vector of arbitrary numbers in zero and one and $\otimes$ denotes the operator indicative of entry-wise multiplication.

The MPA model the "encounter policies" by dividing the optimizer algorithm as 3 equivalent parts. In every phase, the candidate solution updates the location vector in relation to "fittest solution." Thus, the $\overrightarrow{Elite}$ vector is key to MPA. The final stages of MPA are given in the following:

Phase I $\left(iter < \frac{1}{3}MaxIt\right)$

The first third of iteration methods the higher velocity ratio scenario ($\gamma \geq 10$), whereas the prey moves quicker than predator:

$$\overrightarrow{stepsize} = \overrightarrow{R_B} \otimes \left(\overrightarrow{Elite(t)} - \overrightarrow{R_B} \otimes \vec{X}(t)\right), i = 1, \dots, PopSize \quad (15)$$

$$\vec{X}(t+1) = \vec{X}(t) + P.\overrightarrow{rand} \otimes \left(\overrightarrow{stepsize}\right)$$

In Eq. (7), $\vec{X}(t+1)$ denotes the upgraded solution vector, P indicates the constant scaling factor, and $R_B \rightarrow$ shows a vector of randomly created value based on Brownian motion. The $\overrightarrow{R_B}$ vector is multiplied using $\vec{X}(t)$ (marine prey) because fast prey displays random movement.

Phase II $\left(\frac{1}{3}MaxIt < it < 32MaxIt\right)$

MPA map the behaviors of a marine predator as per the unit velocity ratio condition ($\gamma = 1$). Observation shows that both prey and predators are looking for food, and thus, they adopt the opposite strategy for everyone. It can be modeled with half the prey population upgrading the position based on the Lévy approach and the other half based on Brownian motion, viz., 50% exploitation and 50% exploration.

$$\overrightarrow{stepsize} = \overrightarrow{R_L} \otimes (\overrightarrow{Elite(t)} - \overrightarrow{R_L} \otimes \vec{X}(t)) , i = 1, \dots, PopSize/2 \quad (16)$$

$$\vec{X}(t+1) = \vec{X}(t) + P.\overrightarrow{rand} \otimes (\overrightarrow{stepsize})$$

In Eq. (16), $\overrightarrow{R_L}$ shows the vector of arbitrary numbers due to the Lévy distribution:

$$\overrightarrow{stepsize} = \overrightarrow{R_B} \otimes \left(\overrightarrow{R_B} \otimes \overrightarrow{Elite(t)} - \vec{X}(t)\right), i = PopSize/2, \dots, PopSize \quad (17)$$

$$\vec{X}(t+1) = \vec{X}(t) + P.CP \otimes (\overrightarrow{stepsize})$$

$$CF = \left(1 - \frac{t}{\text{Max } It}\right)^{\left(\frac{2t}{\text{Max } lt}\right)} \quad (18)$$

Where CF denotes the control factor, which dictates the step size length of predator movement.

Phase III $\left(\frac{2t}{\text{Max } lt}\right)\left(iter > \frac{2}{3} MaxIt\right)$

The last third of MPA iterations simulated the low-velocity ratio condition ($\gamma \leq 0.1$) and acted as pure exploitation stage.

$$\overrightarrow{stepsize} = \overrightarrow{R_L} \otimes \left(\overrightarrow{R_L} \otimes \overrightarrow{Elite(t)} - \vec{X}(t)\right), i = 1, \dots, PopSize \quad (19)$$

$$\vec{X}(t+1) = \overrightarrow{Elite(t)} + P.CF \otimes (\overrightarrow{stepsize})$$

In this phase, searching agent updates the position around "top predator" and then explores the searching space for 2/3 of iterations, and the refinement of solutions is performed.

Eventually, the population of candidate solution allows updating the location vectors as per a predetermined probability factor allotted by the user:

$$\vec{X}(t+1) = \begin{cases} \vec{X}(t) + CP\left[\overrightarrow{X_L} \rightarrow + \overrightarrow{rand} \otimes \left(\vec{X} - \overrightarrow{X_L}\right)\right] \otimes \vec{U}, r \leq PADs \\ \vec{X}(t) + CF[FADs(1-r) + r]\left(\overrightarrow{X_{rand1}(t)} - \overrightarrow{X_{rand2}(t)}\right), r > FADs \end{cases} \quad (20)$$

In Eq. (20), $FADs$ denote the user-determined probability aspect, $\vec{U}$ indicates the binary vector, r signifies the arbitrary number lies among zero and one, and $rand1$ and $rand2$ denote the arbitrarily prey 1 and chosen prey 2.

Fitness choice is a major issue in the MPAs. An encoding performance can be utilized to assess the good solution of candidate. At present, the acc value is the primary condition mainly applied for scheming a FF.

$$Fitness = \max(P) \quad (21)$$

$$P = \frac{TP}{TP + FP} \quad (22)$$

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

4. Results and Discussion

The pedestrian detection outcomes of the MPADLB-PDC approach are verified on the INRIA Database [24]. It holds 6817 instances with 2 classes, as provided in Table 1. Fig. 3 displays the sample images. Fig. 4 illustrates the original and detected images.

Table 1: Details of database

Classes	No. of Images
Pedestrian (PD)	3578
Non-Pedestrian (NPD)	3239
Total Images	6817

Figure 3. Sample Images

Figure 4. a) Original Images b) Detected Images

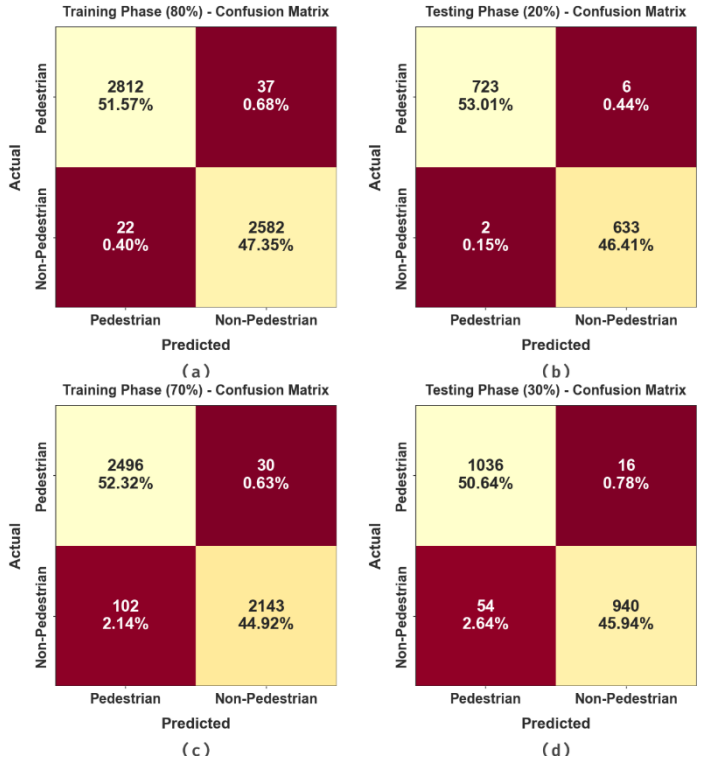

Figure 5. Confusion matrices of MPADLB-PDC system (a-b) 80:20 of TRAP/TSP and (c-d) 70:30 of TRAP/TESP

The confusion matrices of the MPADLB-PDC system on the pedestrian detection procedure are established in Fig. 5. On TRAP 80%, the MPADLB-PDC system classifies 2812 instances into PD class and 2582 samples into NPD class. Also, on 20% of TESP, the MPADLB-PDC approach classifies 723 samples into PD class and 633 samples into NPD class. Additionally, on 70% of TRAP, the MPADLB-PDC system classifies 2496 instances into PD class and 2143 samples into NPD class. At last, on 30% of TESP, the MPADLB-PDC method classifies 1036 instances into PD class and 940 instances into NPD class.

The entire pedestrian detection outcomes of the MPADLB-PDC technique are illustrated in Table 2 and Fig. 6. The simulation outcome stated that the MPADLB-PDC system reached an effective pedestrian identification process. For instance, on TRAP 80%, the MPADLB-PDC technique gains average $accu_y$ of 98.93%, $prec_n$ of 98.91%, $reca_l$ of 98.93%, F_{score} of 98.92%, MCC of 97.83%, and $G_{measure}$ of 98.92%. Simultaneously, on TESP 20%, the MPADLB-PDC system gets average $accu_y$ of 99.43%, $prec_n$ of 99.39%, $reca_l$ of 99.43%, F_{score} of 99.41%, MCC of 98.82%, and $G_{measure}$ of 99.41%.

Table 2: Pedestrian detection of MPADLB-PDC system with distinct measures

Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	MCC	$G_{Measure}$
TRAP (80%)						
PD	98.70	99.22	98.70	98.96	97.83	98.96
NPD	99.16	98.59	99.16	98.87	97.83	98.87
Average	98.93	98.91	98.93	98.92	97.83	98.92
TESP (20%)						
PD	99.18	99.72	99.18	99.45	98.82	99.45
NPD	99.69	99.06	99.69	99.37	98.82	99.37
Average	99.43	99.39	99.43	99.41	98.82	99.41
TRAP (70%)						
PD	98.81	96.07	98.81	97.42	94.48	97.43
NPD	95.46	98.62	95.46	97.01	94.48	97.03
Average	97.13	97.35	97.13	97.22	94.48	97.23

TESP (30%)						
PD	98.48	95.05	98.48	96.73	93.21	96.75
NPD	94.57	98.33	94.57	96.41	93.21	96.43
Average	96.52	96.69	96.52	96.57	93.21	96.59

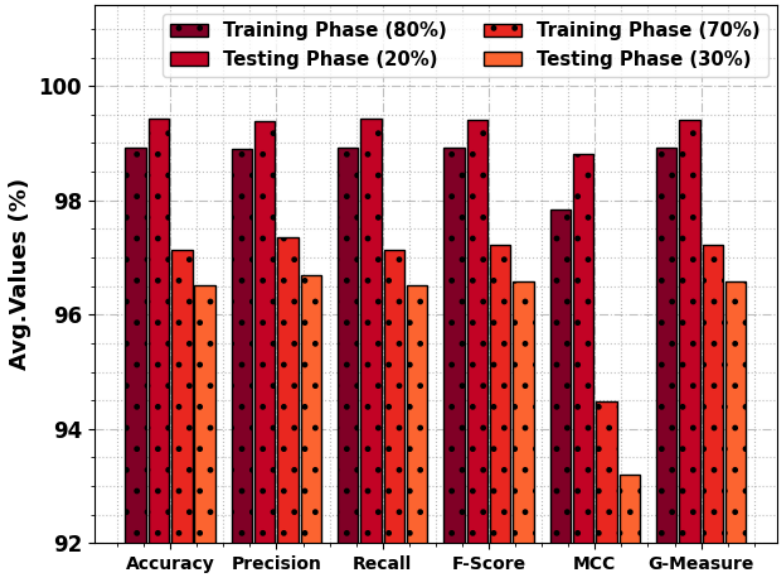

Figure 6. Average outcome of MPADLB-PDC system with distinct measures

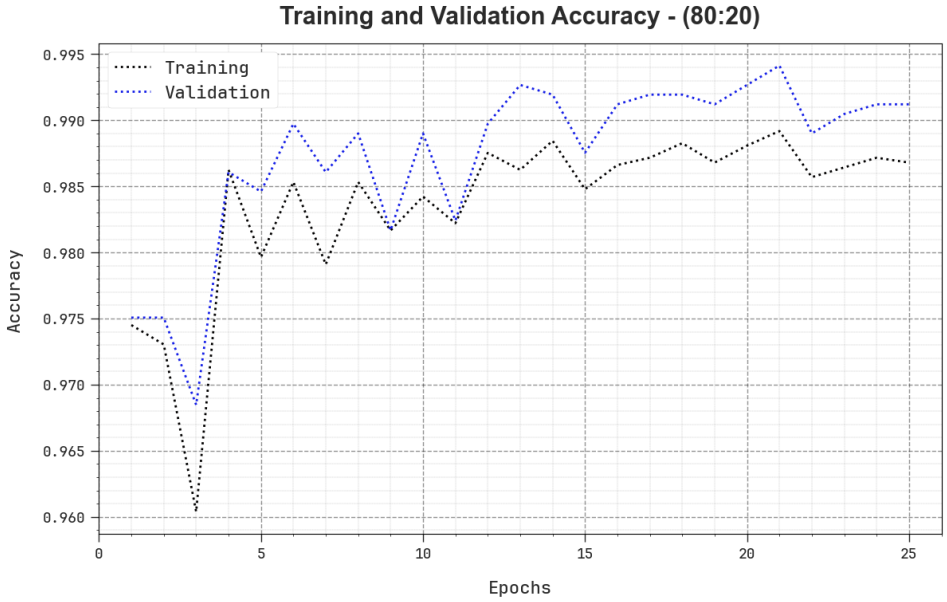

Figure 7. Acc curve of the MPADLB-PDC system on 80%TRAP:20%TESP

Fig. 7 scrutinizes the acc of the MPADLB-PDC system in the TRA and VLA procedure on 80%TRAP:20%TESP. The result inferred that the MPADLB-PDC approach gains higher acc values over enhanced epochs. Besides, the maximum VLA acc over TRA acc reveals that the MPADLB-PDC algorithm learns capably on 80%TRAP:20%TESP.

The loss curve of the MPADLB-PDC approach at the time of TRA and VLA is displayed on 80%TRAP:20%TESP in Fig. 8. The outcome implied that the MPADLB-PDC algorithm attains nearby values of TRA and VLA loss. It could be clear that the MPADLB-PDC approach learns capably on 80%TRAP:20%TESP.

Figure 8. Loss curve of the MPADLB-PDC system on 80%TRAP:20%TESP

The pedestrian detection outcomes of the MPADLB-PDC method undergo comparison with recent DL models in Table 3 and Fig. 9 [25]. From the outcomes, it can be observed that the MPADLB-PDC algorithm exhibits outperforming outcomes over other approaches. With respect to $accu_y$,the MPADLB-PDC technique, offers increased $accu_y$ of 99.43%, while the MMPD-MDCNN, OVGG-16, VGG-16, GAN, RNN, and HMPD models attain decreased $accu_y$ of 99.4%, 98.9%, 96.9%, 97.41%, 97.16%, and 90.74% respectively. Simultaneously, based on $prec_n$, the MPADLB-PDC method offers enhanced $prec_n$ of 99.39%, while the MMPD-MDCNN, OVGG-16, VGG-16, GAN, RNN, and HMPD models realize minimal $prec_n$ of 99.43%, 98.92%, 97.45%, 97.59%, 96.62%, and 89.7% correspondingly. Concurrently, interms of $reca_l$, the MPADLB-PDC system offers superior $reca_l$ of 99.43%, while the MMPD-MDCNN, OVGG-16, VGG-16, GAN, RNN, and HMPD systems achieve lower $reca_l$ of 99.4%, 98.76%, 97.05%, 97.03%, 95.46%, and 91.82% respectively.

Table 3: Comparative outcome of MPADLB-PDC approach with other DL systems

Methods	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}
MPADLB-PDC	99.43	99.39	99.43	99.41
MMPD-MDCNN	99.4	99.43	99.4	99.41
OVGG-16	98.9	98.92	98.76	98.68
VGG-16	96.9	97.45	97.05	96.78
GAN Algorithm	97.41	97.59	97.03	95.83
RNN Algorithm	97.16	96.62	95.46	96.28
HMPD Model	90.74	89.7	91.82	90.93

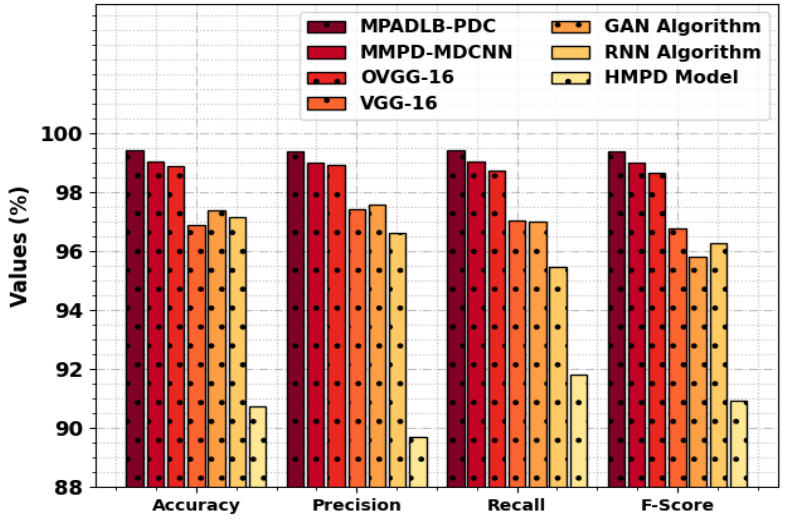

Figure 9. Comparative outcome of MPADLB-PDC algorithm with other DL systems

Finally, based on F_{score}, the MPADLB-PDC approach provides higher F_{score} of 99.41%, while the MMPD-MDCNN, OVGG-16, VGG-16, GAN, RNN, and HMPD methodologies accomplish decreased F_{score} of 99.41%, 98.68%, 96.78%, 95.83%, 96.28%, and 90.93% correspondingly. Consequently, the MPADLB-PDC system exhibited maximum results on the pedestrian detection method.

5. Conclusion

In this manuscript, we have presented an automated pedestrian detection model named the MPADLB-PDC technique, which mainly focuses on the accurate recognition and classification of pedestrians. Primarily, the MPADLB-PDC technique involves two major processes, namely object detection and classification. In the first stage, the MPADLB-PDC technique uses an improved YOLOv7 object detector for the recognition of the objects in the frame. Next, in the second stage, the ensemble classifier comprises three classifiers such as deep feed-forward neural network (DFFNN), extreme learning machine (ELM), and long short-term memory (LSTM). To improve the recognition performance of the ensemble classifier, the MPA is used to optimally select the parameters related to it. The simulation outcome of the MPADLB-PDC approach was validated on pedestrian database, and the outcomes can be studied in terms of various aspects. The simulation values validated the better outcome of the MPADLB-PDC algorithm compared to other approaches.

Funding: "This research received no external funding"

Conflicts of Interest: "The authors declare no conflict of interest."

References

[1] K. V. Kumar, E. L. Lydia, A. K. Dutta, V. S. Parvathy, G. Ramasamy, I. Pustokhina, and D. A. Pustokhin, "Deep Learning Enabled Object Detection and Tracking Model for Big Data Environment," *CMC-Computers, Materials & Continua*, vol. 73, no. 2, pp. 2541–2554, 2022.

[2] M. Jayasuriya, G. Dissanayake, R. Ranasinghe, and N. Gandhi, "Leveraging Deep Learning Based Object Detection for Localising Autonomous Personal Mobility Devices in Sparse Maps," in *Proc. 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 4081–4086.

[3] A. Syed, "Forecasting Pedestrian Trajectory Using Deep Learning," Ph.D. dissertation, Univ. of Nevada, Las Vegas, 2021.

[4] B. B. Elallid, S. E. Hamdani, N. Benamar, and N. Mrani, "Deep learning-based modeling of pedestrian perception and decision-making in refuge island for autonomous driving," in *Computational Intelligence in Recent Communication Networks*, Cham: Springer International Publishing, 2022, pp. 135–146.

[5] O. Angah and A. Y. Chen, "Tracking multiple construction workers through deep learning and the gradient-based method with re-matching based on multi-object tracking acc," *Automation in Construction*, vol. 119, p. 103308, 2020.

[6] F. Sezgin, D. Vriesman, P. Held, A. Zimmer, and T. Brandmeier, "A Deep Learning Approach for Pedestrian Behavior Interpretation Based on Radar Point Clouds," in *Proc. 2021 18th European Radar Conference (EuRAD)*, 2022, pp. 66–69.

[7] X. Wu, D. Sahoo, and S. C. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, 2020.

[8] B. S. Murugan, M. Elhoseny, K. Shankar, and J. Uthayakumar, "Region-based scalable smart system for anomaly detection in pedestrian walkways," *Computers & Electrical Engineering*, vol. 75, pp. 146–160, 2019.

[9] A. Hamadi, "Real-Time Multi-Object Tracking Using Deep Learning," 2021.

[10] D. Kido, T. Fukuda, and N. Yabuki, "Diminished reality system with real-time object detection using deep learning for onsite landscape simulation during redevelopment," *Environmental Modelling & Software*, vol. 131, p. 104759, 2020.

[11] A. Li, S. Sun, Z. Zhang, M. Feng, C. Wu, and W. Li, "A Multi-Scale Traffic Object Detection Algorithm for Road Scenes Based on Improved YOLOv5," *Electronics*, vol. 12, no. 4, p. 878, 2023.

[12] P. K. Y. Wong, H. Luo, M. Wang, P. H. Leung, and J. C. Cheng, "Recognition of pedestrian trajectories and attributes with computer vision and deep learning techniques," *Advanced Engineering Informatics*, vol. 49, p. 101356, 2021.

[13] M. S. Ansarnia, E. Tisserand, P. Schweitzer, M. A. Zidane, and Y. Berviller, "Contextual detection of pedestrians and vehicles in orthophotography by fusion of deep learning algorithms," *Sensors*, vol. 22, no. 4, p. 1381, 2022.

[14] J. Kolluri and R. Das, "Intelligent multimodal pedestrian detection using hybrid metaheuristic optimization with deep learning model," *Image and Vision Computing*, p. 104628, 2023.

[15] P. Sharma, S. Gupta, S. Vyas, and M. Shabaz, "Object detection and recognition using deep learning-based techniques," *IET Communications*, 2022.

[16] S. M. Azimi, M. Kraus, R. Bahmanyar, and P. Reinartz, "Multiple pedestrians and vehicles tracking in aerial imagery using a convolutional neural network," *Remote Sensing*, vol. 13, no. 10, p. 1953, 2021.

[17] M. A. B. Fayyaz and C. Johnson, "Object detection at level crossing using deep learning," *Micromachines*, vol. 11, no. 12, p. 1055, 2020.

[18] B. Zhou, P. Wu, Z. Gu, Z. Wu, and C. Yang, "XDRNet: Deep Learning-based Pedestrian and Vehicle Dead Reckoning Using Smartphones," in *Proc. 2022 IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2022, pp. 1–8.

[19] J. Chen, H. Liu, Y. Zhang, D. Zhang, H. Ouyang, and X. Chen, "A Multiscale Lightweight and Efficient Model Based on YOLOv7: Applied to Citrus Orchard," *Plants*, vol. 11, no. 23, p. 3260, 2022.

[20] B. Amirshahi and S. Lahmiri, "Hybrid deep learning and GARCH-family models for forecasting volatility of cryptocurrencies," *Machine Learning with Applications*, vol. 12, p. 100465, 2023.

[21] Y. Chen, M. Huang, K. Song, and T. Wang, "Prediction of Ship Traffic Flow and Congestion Based on Extreme Learning Machine with Whale Optimization Algorithm and Fuzzy c-Means Clustering," *Journal of Advanced Transportation*, 2023.

[22] B. Amirshahi and S. Lahmiri, "Hybrid deep learning and GARCH-family models for forecasting volatility of cryptocurrencies," *Machine Learning with Applications*, vol. 12, p. 100465, 2023.

[23] R. Bodalal and F. Shuaeib, "Marine Predators Algorithm for Sizing Optimization of Truss Structures with Continuous Variables," *Computation*, vol. 11, no. 5, p. 91, 2023.

[24] [Online]. Available: https://github.com/YoungYoung619/pedestrian-detection-in-hazy-weather

[25] D. K. Jain, X. Zhao, G. González-Almagro, C. Gan, and K. Kotecha, "Multimodal pedestrian detection using metaheuristics with deep convolutional neural network in crowded scenes," *Information Fusion*, vol. 95, pp. 401–414, 2023.