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

Revue d'Intelligence Artificielle
Vol. 38, No. 4, August, 2024, pp. 1159-1172
Journal homepage: http://iieta.org/journals/ria

Real-Time Vehicle Classification Using LSTM Optimized by Oppositional-Based Wild Horse Optimization

Kendagannaswamy Tejaswi*, Ramaiah Krishna Bharathi

Department of Computer Applications, JSS Science and Technology University, Mysuru 570006, India

Corresponding Author Email: tejaaswik@gmail.com

https://doi.org/10.18280/ria.380410

Received: 2 August 2023
Revised: 10 November 2023
Accepted: 29 December 2023
Available online: 23 August 2024

Keywords:
long short-term memory, oppositional based wild horse optimization, ResNet50, vehicle detection and vehicle type classification, Yolov2

ABSTRACT

Classifying vehicles in real time was necessary to manage and plan road traffic and avoid frequent traffic jams, traffic violations, and fatal traffic accidents. However, detecting vehicles at night presents a significant challenge, requiring the classification algorithm to be tested under diverse conditions, such as rainy weather, cloudy weather, low illumination, and others, which makes identifying vehicles a complicated task. This paper detected and classifiess vehicle through YOLO-v2, ResNet50, and an optimally configured Long Short-Term Memory (LSTM). But figuring out the best hyperparameters by trial and error took longer and was more complicated. The research resolved the computational time and complexity by involving Oppositional-based Wild Horse Optimization (OWHO) techniques to identify the optimal hyperparameters for LSTM. The result showed that the proposed technique was better, with an average accuracy of 97.38% in classifying vehicles, which was better than other techniques.

1. INTRODUCTION

Urban and highway traffic analysis and planning tools rely on key elements like vehicle classification (cars, trucks, buses, autos, and two-wheelers) on urban roads and highways, as well as statistical traffic flow estimation [1]. Real-time highway traffic flow monitoring is still a challenging undertaking, nevertheless, in this day of rapidly advancing technology and urbanisation. Traditional approaches, such as human observers, fall short as they are inadequate for vehicle recognition, classification, and generating real-time traffic flow data [2]. Human observers can be error-prone and subjective, leading to inaccurate and inconsistent results. They may also work at a slower pace, be affected by fatigue, and struggle in challenging conditions like bad weather or low-light situations. Additionally, employing human observers for large-scale traffic monitoring can be inefficient and costly, making automation a more practical choice. Inadequate road/highway traffic management leads to traffic law violations, congestion, and accidents. Conventional techniques (e.g., RADAR, LIDAR, RFID, or LASAR) require significant time, money, and effort [3]. The challenge lies in achieving automatic vehicle classification from traffic surveillance camera recordings [4]. Although there are publicly available vehicle datasets, not all are suitable for training traffic surveillance algorithms [5]. The research's key contributions include addressing real-time vehicle classification for traffic management, handling diverse environmental conditions, combining YOLO-v2, ResNet50, and optimized LSTM models, streamlining hyperparameter optimization with OWHO techniques.

Traffic surveillance photographs that are captured have a lesser resolution and are affected by different lighting, weather, and occlusion circumstances [6]. Modern artificial intelligence techniques, especially those based on deep learning and machine learning, are harnessed by online video processing systems [7]. Convolutional neural networks (CNN) - ResNet-50 are utilized in the study for automatic feature extraction. Compared to conventional feature extraction methods, this approach is more reliable and discriminative. Models based on ResNet-50 have the ability to share parameters and have local connectivity. The features from various layers are extracted utilizing the ResNet-50's layered architecture [8]. There is still a lot of room for classification study even if LSTM has dealt with time series issues adequately. Vehicle identification using the conventional LSTM network model alone cannot produce better classification results [9].

LSTM networks are well-suited for handling temporal sequence data like traffic surveillance. Integrating hyperparameter optimization is necessary to adapt the model to the specific challenges and complexities of traffic data, improve its performance, and ensure its ability to generalize effectively to real-world scenarios.

To enhance the LSTM model, the Paper aim to identify optimal hyper parameters. It is possible to think about hyper parameter optimization as an optimization issue where the goal is to choose a value that maximises performance and produces the desired model [10]. The goal of the scientific field of optimization is to maximise or minimise particular objective functions. It is present in almost all disciplines, including economics, text clustering, and pattern recognition. Recent attempts to resolve these issues have been made by academics employing metaheuristics, a novel class of approximation method [11]. The prior swarm-based optimization techniques have some flaws, worries, and issues. This paper analyses the WHO optimizer algorithm, which was motivated by wild horses' social behaviours [12]. Therefore, in order to explain complex problems and functions, this algorithm needs to be enhanced. It is a concept of computational opposition that aims to accelerate the convergence of soft computing algorithms. It is based on the idea of opposed interactions between things. Calculating both the initial outcome and its reverse is required [13].

The following sections are organised as follows: Section 2 illustrates the recent literature reviews in this study context. The subsequent research demonstrates the methodology in section 3, results, and discussion with interpretations in section 4, followed by the conclusion.

2. LITERATURE REVIEW

In a real-world road context, Karungaru et al. [14] suggested utilising an enhanced and updated AlexNet for vehicle recognition and categorization. The vehicle categorization network was expanded to include spatial pyramid pooling. The suggested approach performs better at detecting and classifying vehicles.

Object detection in automatic driving systems and driver assistance systems was suggested by Han et al. [15]. Small vehicle object real-time detection models suffer from low precision and subpar performance. A deep learning system for object detection called YOLO-v2 is offered by them. The analysis of the KITTI dataset and the experiment results show that, without losing detection speed, the model not only improves the accuracy of recognising small vehicle objects but also improves the accuracy of detecting all vehicles, reaching an accuracy of 94%.

Hou et al. [16] introduce a novel online detector for construction vehicles, leveraging the YOLO network's efficiency, high regression rate, and reduced computational demands. The proposed detector achieves an impressive detection accuracy of more than 94.79%, as confirmed through simulation verification. The detector's network structure is built upon the feature extraction capabilities of the Resnet 50 network.

A model for efficient multi-scale vehicle target recognition in traffic scenes is presented by Luo et al. [17]. It combines Faster R-CNN with NAS optimisation and feature enrichment. They propose utilising an image adaptive correction method based on Retinex to improve the quality of the traffic images in the collection. Moreover, the model makes use of feature enrichment, which better understands vehicle targets by integrating multi-layer feature information with the final layer through cross-layer connections.

Şentaş et al. [18] utilised an SVM classifier in conjunction with the tiny YOLO for vehicle detection (VD) and classification. To assess its performance, the developed model was tested on the BIT Vehicle Dataset, focusing on precision and recall measures. The outcomes of the experiment showed that the model could successfully identify different kinds of vehicles in traffic footage that were being streamed in real time. However, a noteworthy limitation in this study was the use of SVM, which being a binary classifier, only allowed for binary classification tasks.

A deep learning-based technique for categorising different kinds of vehicles in intermediate road traffic was presented by Kolukisa et al. [19] They collected 376 vehicle samples and established classifications for light, medium, and heavy vehicles. The study found that the most effective transfer learning approach for vehicle type classification is the soft voting ensemble technique, which combines LSTM, GRU, and VGG16 models. Comparative performance analysis revealed significant improvements in accuracy (92.92%) and f-measure (93.42%) when utilizing the deep learning classifier with the soft voting ensemble technique [19].

A brand-new pseudo LSTM classifier for single image vehicle classification is presented by Rachmadi et al. [20] Pseudo LSTM classifiers, in contrast to conventional ones, use spatially segmented pictures rather than time-series data. By cropping input photographs utilising a two-level spatial pyramid region design, the images are divided. Spatial pyramid features of these divided images are then extracted using parallel convolutional networks. This innovative approach demonstrates the adaptability of LSTM classifiers, which are conventionally utilized for time-dependent data, for handling non-time-dependent data as well.

Autonomous driving relies on recognizing vehicles and pedestrians, but choosing a detection system that balances accuracy, speed, and memory usage is increasingly challenging due to numerous methodologies. Chen et al. [21] examine common object detection architectures (Faster R-CNN, R-FCN, and SSD) and feature extractors (ResNet50, ResNet101, MobileNet V1, MobileNet V2, Inception V2, and Inception ResNet V2). Extensive testing on the widely used KITTI benchmark reveals that Faster R-CNN ResNet50 achieves the best car and pedestrian recognition performance, with an impressive 58% average precision (AP) at a speed of 8.6 FPS.

In order to address the dependability redundancy allocation problem for series-parallel systems, AL-Saati [22] presents the WHO method. The efficiency of this technique was tested using four well-known numerical cases. The outcomes are contrasted with other algorithms such as the simplified swarm algorithm and the competitive attraction-repulsion algorithm.

By using the grasshopper optimisation algorithm, Barun Mandal et al. [23] suggested a novel approach to deal with the dynamic economic load dispatch problem in power systems. However, this algorithm, like others, suffered from issues such as premature convergence and slow convergence rate. To tackle these challenges, they integrated oppositional based learning with the grasshopper optimization algorithm to enhance its convergence mobility. The opposing based chaotic grasshopper optimisation method was shown to be the best efficient solution for the dynamic economic load dispatch problem.

Numerous investigators have proposed dissimilar methods utilizing diverse source inputs for detecting vehicles in day and night vision scenarios. Evaluating the classification algorithm under various conditions, including challenging scenarios like rainy, snowy, and low illumination conditions, is necessary. Arora and Kumar [24] provided a comprehensive review of studies on VD during both day and night. They discussed about dissimilar approaches for detecting vehicles as well as the function that intelligent transportation systems have in identifying and detecting vehicles. The paper also presents a concise summary of reported approaches for identifying dissimilar vehicle types in various settings and the challenges faced by other researchers in this field.

Lin et al. [25] introduce AugGAN, a data augmenter based on Generative Adversarial Networks (GANs), capable of transforming on-road driving images into a desired domain while preserving image objects effectively. They quantitatively assess various approaches by training Faster R-CNN and YOLO using datasets generated from the transformed results. Utilising the suggested AugGAN model significantly improves object detection accuracies, according to the experimental results.

Bell et al. [26] propose a real-time VD system designed for nighttime conditions. The system utilizes earlier complex light patterns in the image to detect automobiles. They developed a machine learning system based on a grid of foveal classifiers, where each classifier processes the same global picture description (one descriptor per image). However, each classifier is trained to predict a distinct outcome based on its grid position and the location of the ground truth vehicle. This approach allows for simple point-based annotations during training, reducing the time and cost required for dataset creation. Experimental results demonstrate the effectiveness of this strategy on a newly built nighttime database with point-based annotations.

The literature review reveals key findings in the field of vehicle detection and classification. These findings include the effectiveness of enhanced AlexNet for vehicle recognition, improved small vehicle detection with YOLO-v2, efficient construction vehicle detection using YOLO, and multi-scale vehicle target detection techniques. Additionally, the review highlights the use of small YOLO for vehicle detection and classification, effective transfer learning for vehicle type classification, novel pseudo LSTM classifiers for single image vehicle classification, optimal object detection architectures, and methods for addressing dependability redundancy allocation. Furthermore, it discusses data augmentation with AugGAN, real-time vehicle detection for nighttime conditions, and emphasizes the importance of evaluating algorithms in various challenging scenarios. These insights contribute to advancements in vehicle detection and classification techniques for diverse real-world applications.

Research gap

- Limited comparison of the proposed OWHO technique with other hyperparameter optimization approaches for LSTM in the context of vehicle classification in real-time traffic monitoring systems.
- Lack of attention to the specific characteristics of LSTMs that make them unique for vehicle classification in real-time traffic monitoring systems, such as handling sequential data and the use of gates.
- Lack of attention to scalability of OWHO for large datasets and architectures in real-time traffic monitoring systems.
- Lack of attention to the interpretability of the LSTM models for vehicle classification in real-time traffic monitoring systems.
- Lack of attention to the robustness of LSTM models to different types of variations in the input data in real-time traffic monitoring systems, like low illumination, rainy, and cloudy weather.
- Limited examination of the ability of the proposed OWHO method to be implemented in real-world scenarios and to improve the performance of real-time traffic monitoring systems and reduce traffic congestion and accidents.

The following are this paper's main contributions:

- The objective of this study is to develop a methodology for identifying and classifying vehicles in real-time, with a focus on challenging conditions such as low illumination and poor weather.
- The proposed approach utilizes a combination of YOLO-v2, ResNet50, and an optimally configured LSTM network for VD and classification.
- Additionally, the study explores the use of OWHO techniques for identifying the optimal hyperparameters of the LSTM network, in order to increase performance and decrease computational time and complexity.
- The proposed method's effectiveness is assessed by utilizing a real-world traffic video dataset and comparing the outcomes with those obtained from other techniques.

3. PROPOSED METHODOLOGY

A precise and effective VD method is crucial because it is the first stage in the process of classifying a vehicle type. The accuracy of traditional VD and classification techniques decreases dramatically due to the backdrop, changing lighting, climatic conditions, and varying vehicle sizes in a frame. The motion-based VD system may identify moving cars in video frames. The method, however, is unable to recognise items in a single image and fails to recognise stationary objects. Furthermore, feature-based VD is not universal because finding suitable features to describe the thing is challenging. This research involves a popular real-time object detection algorithm named YOLOv2 for object detection. ResNet-50, a CNN with 50 layers, is then exploited in the study for the feature extraction procedure. The extracted features from ResNet50 are used as input for LSTM in order to classify vehicle types. This research includes four classes: Light Moving Vehicle (LMV), Heavy Moving Vehicle (HMV), Auto, and Two-Wheeler (TW). Furthermore, the research intends to configure the LSTM’s hyperparameters suitable for classifying vehicle types. Deciding the optimal hyperparameters is crucial in determining the overall LSTM performance. Because they have a tremendous influence on the final output, it is necessary to consider optimal hyperparameters carefully. However, it is complicated and time-consuming to identify the optimal hyperparameters through a manual or trial and error process. Therefore, the research integrates WHO to resolve the computational complexity while configuring the ANN architecture. Fig. 1 shows the overall research methodology starting from video to frame conversions and getting into object detection using Yolo-v2, then feature extraction using ResNet50 and finally getting into classification with LSTM.

3.1 Dataset description

This study employs real-time vehicle surveillance videos collected from the Surveillance Camera of the Transport Department in Mysuru, India. A total of 36 videos were selected for this research, which represent daytime lighting, cloudy, and nighttime lighting conditions. Out of the 36 real-time vehicle surveillance videos, 30 were utilized for training purposes, with 11 representing daytime lighting conditions, 14 representing cloudy conditions, and 5 representing night-time lighting conditions. Each video has duration of 3 seconds and contains 100 frames, resulting in a total of 1100 frames for the daytime lighting condition, 1400 frames for the cloudy condition, and 500 frames for the night-time lighting condition. The remaining 6 videos were reserved for testing purposes. Figure 1 illustrates overall proposed method.

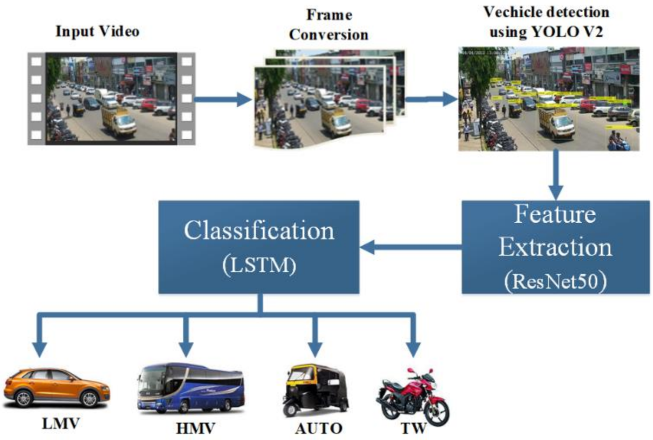

Input Video
Frame Conversion
Vechicle detection using YOLO V2
Feature Extraction (ResNet50)
Classification (LSTM)
LMV
HMV
AUTO
TW

Figure 1. Overall research methodology

3.2 Yolo-v2

The YOLO-v2 object detector is used in this research as a methodology as a part of identifying and classifying vehicles in real-time. The role of YOLO-v2 in this research is to detect and classify objects in the videos, which provides valuable data for the analysis and interpretation of road traffic conditions also helps in improving the road safety and traffic management. The YOLO-v2 process was performed on a total of 3000 frames, comprising 1100 frames representing daytime lighting conditions, 1400 frames representing cloudy conditions, and 500 frames representing night-time lighting conditions. These frames were utilized for the detection of vehicles. The YOLOv2 model can process frames under different lighting conditions, such as cloudy conditions, night-time lighting conditions, and day lighting conditions, in the same way as it processes frames under any other lighting conditions. For each frame, the YOLOv2 model processes the image in a single forward pass, generating predictions for the presence of vehicle and their bounding boxes. The model outputs the coordinates of the bounding boxes, the class probabilities for each box, and the confidence scores for each prediction. The raw predictions from the YOLOv2 model need to be post-processed to filter out low-confidence predictions, and to merge overlapping bounding boxes. The processed frames can be displayed with the detected objects and their bounding boxes drawn on them. Under cloudy conditions, the input frames may have lower overall brightness, making it harder for the model to differentiate between objects and the background. During night-time lighting conditions, the input frames can have very low overall brightness, and may have areas of high contrast, such as streetlights. This can make it harder for the model to identify objects, especially if the objects have similar colors or textures to the background. Under day lighting conditions, the input frames will have higher overall brightness and more consistent lighting, making it easier for the model to identify objects and their bounding boxes. YOLOv2 is capable of detecting vehicle that is partially occluded by other objects or by the boundaries of the image. This makes it a good choice for VD under different lighting conditions, as objects may be partially obscured by shadows or other objects in the scene. YOLOv2 is an end-to-end learning system that predicts object locations and class probabilities in a single forward pass. This makes it a good choice for object detection under different lighting conditions, as it does not require separate detection and classification steps. By using YOLOv2 for VD under different lighting conditions, the advantage is real-time processing speed, single pass detection, end-to-end learning, and robustness to partial occlusions, and improved accuracy. The following Figure 2 demonstrates the working flow of Yolo-v2.

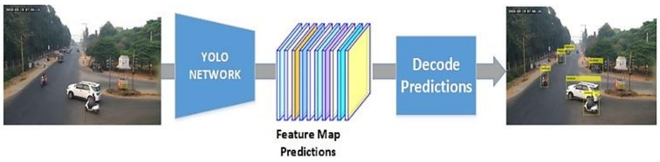


Figure 2. YOLO-v2 process flow

3.3 Convolution neural network (CNN)

CNN, neural network architecture, excels in image recognition [27] and various other visual data processing tasks. ResNet50 is a specific architecture of CNN that was introduced in a 2015 paper by researchers at Microsoft. ResNet50 uses a technique called "residual connections" to allow the network to learn more effectively even as it becomes deeper. This allows ResNet50 to achieve better performance than other CNN architectures on image recognition tasks. In this study, ResNet 50 was used for the feature extraction process in order to identify and classify vehicles in real time. ResNet 50, a deep residual network, was utilized for feature extraction due to its ability to effectively extract features from images, even in low-light conditions. The ResNet 50 model architecture is able to learn deep residual representations by using identity mappings, which allows for the network to be trained with much deeper architectures than traditional approaches. This helps to improve the accuracy of the feature extraction process and allows for more accurate vehicle classification.

3.4 Recurrent neural network (RNN)

LSTM is a type of RNN that is able to capture long-term dependencies in sequential data. LSTMs are able to overcome the vanishing gradient problem that occurs in traditional RNNs by introducing a memory cell, input gate, forget gate, and output gate. These gates allow LSTMs to selectively store or discard information in the memory cell, which allows them to maintain a useful representation of the past for long periods of time. LSTM is a variation of the RNN architecture which is able to learn the context of the data, and capture the dependencies of the input elements over time. While RNN's are designed to capture the sequence information and LSTM are designed to capture the long-term dependencies.

An RNN is an ANN with arbitrary connections between neurons, often fully connected between neighbouring layers. The inputs that the network nodes receive are the current data point x (t) and the hidden state values of the hidden layer in the previous state h (t-1). As a result, by virtue of recurrent connections, inputs at time t have an effect on the network's future outputs. A standard RNN with an input vector $ve=(ve_1, ..., ve_T)$ calculates a hidden vector $hd=(hd_1, ..., hd_T)$ and an output vector $oy=(oy_1, ..., oy_T)$ by iterating Eq. (1) and Eq. (2) over $t=1, \dots, T$.

$$hd(t) = A_Q(Wgt_{(hx)}x^{(t)} + Wgt_{(hh)}hd^{(t-1)} + bi_h) \qquad (1)$$

$$oy^{(t)} = \sigma(Wgt_{(hy)}hd^{(t)} + bi_y) \quad (2)$$

where, bi_y and bi_h are vectors of biases, $Wgt_{(hx)}$ $Wgt_{(hh)}$ and $Wgt_{(hy)}$ are weights matrices of input-hidden layer, hidden-output layer and recurrent connections respectively. A_Q is an activation function [24]. Standard neural networks are trained using the backpropagation via time process over a number of time steps.

3.5 Long term short term memory networks

RNN from the 1980s are where LSTM networks got their start. To solve the disappearing and expanding gradient issues of conventional RNNs, Hochreiter and Schmidhuber created the RNN architecture [28]. Their architecture makes it possible to gather information while in use and employs feedback to keep track of prior network call conditions. In comparison to traditional RNNs, the LSTM model has demonstrated exceptional ability to learn long-range needs in real-world applications. As a result, the LSTM model is used in the majority of cutting-edge applications. The LSTM typically consists of a few memory blocks. Memory cells and gates are found in memory blocks. Information flow is controlled by gates, and memory cells are capable of remembering the network's temporal state through self-connections. A forget gate, an output gate, and an input gate are all included in every memory block. The output gate regulates how cell activations are distributed throughout the rest of the network. The LSTM cell's architecture is depicted in Figures 3 and 4.

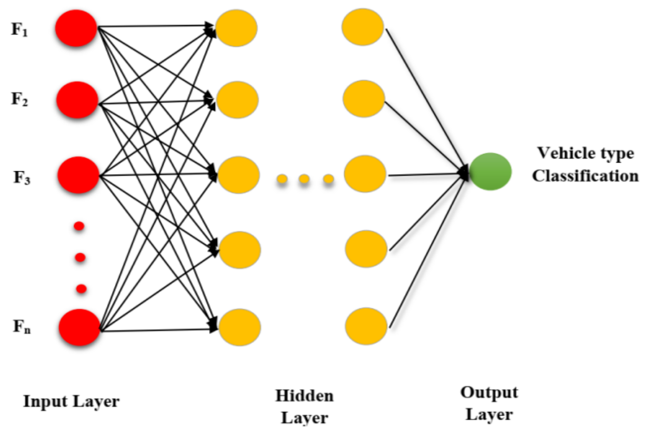


Figure 3. Flowchart for the LSTM's classification of vehicle types

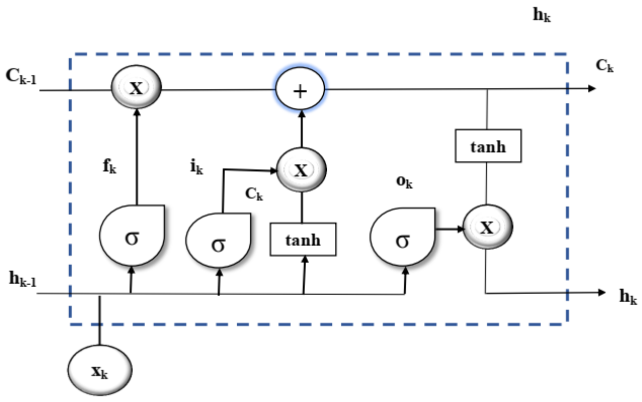


Figure 4. RNN-LSTM layer model

LSTM is used as a classifier in conjunction with YOLO-v2 and ResNet50 for identifying and classifying vehicles in real-time. The LSTM is configured optimally to increase the accuracy of the VD and classification process. Additionally, the research utilizes OWHO techniques to identify the optimal hyperparameters for LSTM, which helps to improve the performance of the overall method. The use of LSTM allows for the detection and classification of vehicles under various factors like Rainy weather, cloudy weather, Low illumination, etc., which can be challenging for other algorithms.

Hyperparameter optimization is essential in LSTM because it allows the model to find the best set of parameters for the specific task and dataset being used. LSTMs have many hyperparameters, like the number of hidden units, the number of layers, and the learning rate, that can greatly impact the performance of the model. By optimizing these hyperparameters, the model can be fine-tuned to achieve better results, such as higher accuracy or lower loss. Without proper optimization, the model may not perform as well or may take longer to converge. Optimization techniques are essential in hyperparameter optimization because they allow for efficient and automated search for the optimal set of hyperparameters. Without optimization techniques, finding the best set of hyperparameters would likely require a significant amount of time and resources, as it would involve manually testing different combinations of hyperparameters.

3.6 Opposition based Wild Horse Optimization (OWHO)

WHO is a metaheuristic optimization algorithm that is inspired by the behavior of wild horses utilized for identifying the hyperparameters for LSTM. It is a population-based optimization algorithm that can be used to solve optimization problems in various fields. WHO uses a group of individuals, called a population, to represent possible solutions to the optimization problem. The algorithm iteratively updates the population by moving the individuals towards better solutions. WHO has been utilized in different applications, like image processing, signal processing, and machine learning. Oppositional-based learning is a technique that involves adding an opposite strategy to the traditional optimization algorithm. In the context of WHO, adding an opposite strategy can help improve performance by allowing the algorithm to explore a wider range of solutions. This can be particularly useful when dealing with complex, high-dimensional problems that have multiple local minima. The advantages of using opposition-based learning include faster convergence, increased robustness, and improved global search capabilities. Additionally, the opposition-based approach can help to mitigate the effects of premature convergence and stagnation, which are common problems in traditional optimization algorithms. Overall, incorporating opposition-based learning into traditional optimization methods like WHO can lead to more efficient and accurate solutions for a wide range of optimization problems.

Horse social structure: horses are separated into territorial and non-territorial groups. Social grouping, bonding and grazing, mating behaviour, leadership hierarchy, and dominance are only a few of the many contrasts between these two types of organisations. In this essay, we'll be concentrating on equine extraterrestrials. Herds of non-terrestrial horses form stable family units known as harems, comprising a stallion, one or more mares, and their young. Stallions are positioned near mares for communication, and mating can occur at any time. Foals start grazing within their first week of life, increasing grazing and reducing rest as they grow older. As foals reach adolescence, they leave their parent groups, and male horses old enough to breed join single groups, while female foals join other family groups. This separation prevents mating between fathers and their offspring. The suggested wild horse optimizer approach involves five primary phases, as shown in Figure 5 of the flowchart.

Establishing the initial population, setting up horse groups, and choosing leaders:

- Horse grazing and mating.
- The leader (stallion) leads and guides the group.
- Leaders are exchanged and chosen.
- Keep the finest solution.

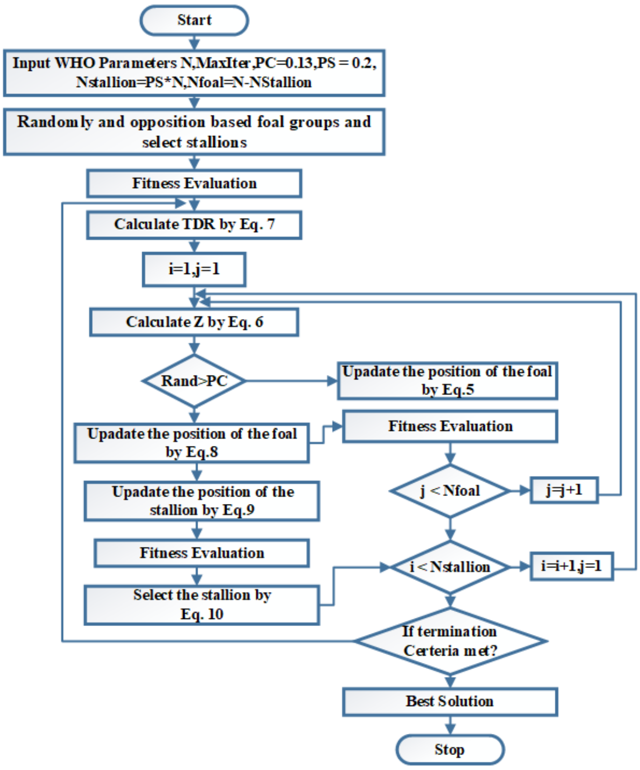

Start
Input WHO Parameters N,MaxIter,PC=0.13,PS = 0.2, Nstallion=PS*N,Nfoal=N-NStallion
Randomly and opposition based foal groups and select stallions
Fitness Evaluation
Calculate TDR by Eq. 7
i=1,j=1
Calculate Z by Eq. 6
Rand>PC
Upadate the position of the foal by Eq.5
Upadate the position of the foal by Eq.8
Fitness Evaluation
Upadate the position of the stallion by Eq.9
j < Nfoal
j=j+1
Fitness Evaluation
i < Nstallion
i=i+1,j=1
Select the stallion by Eq. 10
If termination Certeria met?
Best Solution
Stop

Figure 5. Flow chart of oppositional based wild horse optimization

3.6.1 Creating an initial population

Every optimization algorithm shares the same fundamental structure. The algorithm begins with an initial random population of $(\vec{x})=\{\vec{x_1}, \vec{x_2}, \ldots, \vec{x_n}\}$. This random population is evaluated by the target function multiple times, yielding the target value $(\vec{Ot})=\{Ot_1, Ot_2, \ldots, Ot_n\}$. A set of guidelines for an optimisation strategy was also useful. There is no guarantee that a solution will be found in a single run because population-based optimisation approaches look for the appropriate number of optimisation problems. However, the likelihood of discovering the best global solution rises with enough random solutions and optimization stages (iteration). We first divide this initial population into a number of categories. If N is the population's total size, then G=[N×PS] is the total number of groups. We refer to the PS, which is the percentage of stallions in the overall population, as a control parameter for the suggested approach. The leader G (stallion) and the other members (N-G) are split evenly among the dissimilar groups, according to the number of them. Prior to being chosen by an algorithm based on fitness (the best fitness function) among the group members, group leaders are originally chosen at random.

3.6.2 Opposition based solution

The opposition-based strategy works by creating an "oppositional" solution that is the opposite of the best solution found so far. This oppositional solution is then used as a starting point for the next optimization iteration. The idea behind this is that by exploring the opposite of the best solution, the algorithm can discover new, previously unexplored regions of the solution space that may contain even better solutions. The opposition-based strategy is particularly useful for hyperparameter optimization problems because it allows the algorithm to explore a wide range of possible solutions. This can be especially beneficial when the solution space is complex, highly nonlinear, or has many local optima.

$$OH^{j}_{i,G} = x_i + y_i - H^{j}_{i,G} \tag{3}$$

In Eq. (3), OH represents the opposition-based solution, and H denotes the randomly generated solution with x_i and y_i representing the minimum and maximum values, respectively. Both the randomly generated solution and the opposition-based solution are used in the fitness computation for process evaluation.

3.6.3 Fitness computation

In LSTM models, accuracy serves as the fitness function for hyperparameter optimization, measuring how effectively the model classifies vehicle types based on input data. Typically, accuracy is computed by comparing the model's predictions against the true outputs on a test dataset. The OWHO algorithm utilizes accuracy as a guide to find optimal hyperparameters that yield the highest accuracy on the test set. This process involves iteratively testing various hyperparameter combinations and adjusting them to enhance the model's accuracy.

$$Accuracy = \frac{T_rP + T_rN}{T_rP + T_rN + F_aP + F_aN} \tag{4}$$

In Eq. (4), T_rP refers True Positive; T_rN refers True Negative; F_aP refers False Positive; F_aN refers False Negative.

3.6.4 Grazing behaviour

Foals often graze about their group for the majority of their time, as was indicated in the preceding section. To carry out grazing behaviour, we consider the stallion to be the centre of the grazing area. Group members then search the region around the stallion (graze). To imitate grazing behaviour, we presented Eq. (5), which leads group members to move and search in a dissimilar radius around the leader.

$$\begin{aligned}\overline{X}^{j}_{i,G} &= 2F\cos(2\pi RF)\times \\ &(Stallion^{j} - H^{j}_{i,G}) + Stallion^{j}\end{aligned} \tag{5}$$

where, $H^{j}_{i,G}$ is the current position of the group member (foal or mare), $Stallion^{j}$ is the position of the stallion (group leader), F is an adaptive mechanism calculated by Eq. (6), R is a uniform random number in the range [−2, 2] that causes The grazing of horses at dissimilarangles (360 degrees) of group leader, π is the same as the pi number equal to 3.14,The COS function by combining π and R causes the movement in dissimilar radius, and finally $\overline{X}_{i,G}^{j}$ is the new position of the group member when grazing:

$$\begin{aligned} &P_v = \vec{R}_1 < TDR; \mathrm{IRV} = (P_v == 0); \\ &F = R_2 \Theta \mathrm{IRV} + \vec{R}_3 \, \Theta(\sim \mathrm{IRV}), \end{aligned} \tag{6}$$

where, P_v is a vector consisting of 0 and 1 equal to the dimensions of the problem, $\overrightarrow{R_1}$ and $\overrightarrow{R_3}$ are random vectors with uniform distribution in the range [0, 1], R_2 is a random number wit uniform distribution in the limit [0, 1], IRV indexes of the random vector $\vec{R}_1$ returns that satisfy the condition (P_v==0). TDR is an adaptive parameter that initiates with a value of 1 and progressively decreases as the algorithm is executed following Eq. (7) until it reaches a value of 0 at the end of the algorithm's execution.

$$TDR = 1 - iter \times \left(\frac{1}{M_{iter}} \right) \tag{7}$$

where, M_{iter} represents the maximum number of times the algorithm can be executed, while iter indicates the current run.

3.6.5 Horse mating behaviour

Horses separate their foals from the group and breed them, which is one of their distinctive behaviours in comparison to other animals. Female foals join another family group once they reach puberty in order to find a partner, while male foals leave the family group before they reach adulthood and join the group of single horses. To stop the father from mating with the daughter or other family members, he is leaving. Here's how we put this behaviour into practise: A foal joins a temporary group after leaving group I, while another departs group j.Since these two foals have no familial ties, we'll presume they are both male and female and can mate once they reach adolescence. The offspring is required to leave the temporary group and join another group, denoted as group k. This cycle of emigration, mating, and reproduction is a common occurrence in all equine species. To model the behavior of horses leaving and mating, Eq. (8) is introduced, which corresponds to the Crossover operator of the mean type.

$$\begin{aligned} &H_{G,k}^{p} = Crossover\left(H_{G,i}^{q}, H_{G,j}^{z}\right) \\ &i \neq j \neq k, p = q = end, \\ &\qquad \text{Crossover=Mean} \end{aligned} \tag{8}$$

In the context of the equation, $H_{G,k}^{p}$ represents the position of horse p from group k, which leaves the group to create space for a new horse whose parents are horses from group i and j, having reached puberty and mated to reproduce. These horses have no family relationship. Additionally, $H_{G,i}^{q}$ indicates the position of foal q from group i, which also leaves the group and, after reaching puberty, mates with the horse z in position $H_{G,j}^{s}$ who then departs from group j.

3.6.6 Group leadership

The group leader is responsible for directing the group to the proper location. This appropriate location is what we refer to as the water hole. This water hole must be approached by the group. In a same manner, other groups approach this water hole.In order for the domination group to use this water hole, leaders compete for it. No other groups may utilise the water hole until the dominating group has left. The group leaders must direct their members to the watering hole, use it if their group is the dominant one, and migrate away from it if another group is. Eq. (9) is suggested for this technique and distance.

$$\overline{Stallion_{G_i}} = \begin{cases} 2F\cos(2\pi RF) \times (WH - Stallion_{G_i}) + WH & if\ R_3 > 0.5 \\ 2F\cos(2\pi RF) \times (WH - Stallion_{G_i}) - WH & if\ R_3 \leq 0.5 \end{cases} \tag{9}$$

$$Stallion_{G_i} = \begin{cases} H_{G,i} & if\ \cos t(H_{G_i}) < \cos t(Stallion_{G_i}) \\ Stallion_{G_i} & if\ \cos t(H_{G_i}) > \cos t(Stallion_{G_i}) \end{cases} \tag{10}$$

where, $\overline{\text{stallıon}}_{\text{Gi}}$ is the next position of the leader of the i group, WH is the position of the water hole, $stallion_{Gi}$ is the current position of the leader of the i group, F is an adaptive mechanism calculated by Eq. (10), R is a uniform random number in the range [−2, 2], π is the same as pi number equal to 3.14.

3.6.7 Exchange and leader selection

To retain the randomization of the method, we initially choose the leaders at random. Leaders are chosen later in the process based on fitness. Eq. (10) will alter the dynamic between the group leader and corresponding member if one of the group members is more physically fit than the group leader.

4. RESULTS AND DISCUSSION

The research considers different classification approach namely ResNet-50, Support Vector Machine (SVM), LSTM, and then integrating Particle Swarm Optimization (PSO) and WHO optimization techniques for enhancing the performance of traditional LSTM by predicting the optimal hyperparameters. However, the average vehicle categorization accuracy of this conventional approach is poor, which is unsuitable for traffic flow monitoring. Consequently, the research integrates OWHO in finding optimal hyperparameters for LSTM that accomplished effective traffic monitoring systems by detecting vehicle and type classification. The performance of the proposed approach evaluates through various standard measures like sensitivity, specificity, accuracy, Positive Predictive Value (PPV), Negative Predictive Value (NPV), False Positive Rate (FPR), False Negative Rate (FNR), and False Discovery Rate (FDR). These performance measures are evaluated with the aid of TP, FP, TN and FN and these terminologies are utilize to derived the vehicles in the class.

For instance, if the terminologies are applied for computing LMV.

True Positive: LMV correctly identified as LMV vehicle

False Positive: Other vehicle in the class incorrectly identified as LMV

True Negative: Other vehicle in the class correctly identified as other vehicle

False Negative: LMV incorrectly identified as other vehicle in the class

The research integrates Yolo-v2 for object detection and the LSTM-OWHO for classification; the Figures 6-8 exhibits the performance of proposed approach for sample frame for dissimilar lighting conditions such as day-time, cloudy and the night-time conditions. It is evident from the following results that the proposed approach (i.e) OWHO utilize to configure the LSTM hyperparameters having better performance compares with the other employed techniques in all evaluated measures.

Figures 9-11 illustrate the performance of the employed techniques concerning standard measures.

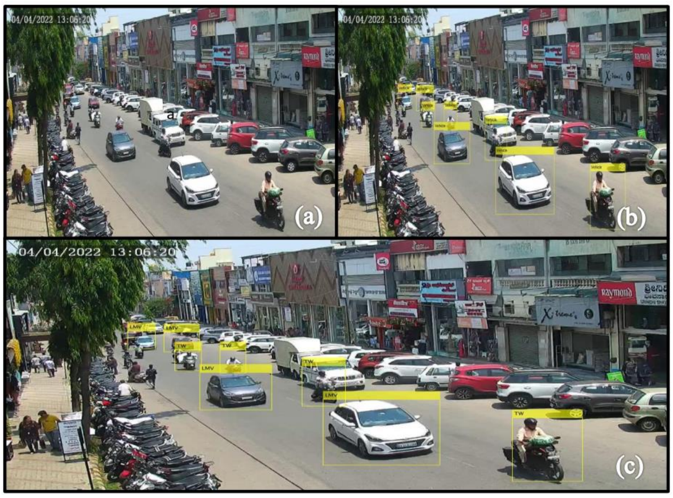


Figure 6. (a) Sample input video-frame; (b) object detection using Yolo-v2 and (c) vehicle type classification from LSTM-OWHO for day-time condition

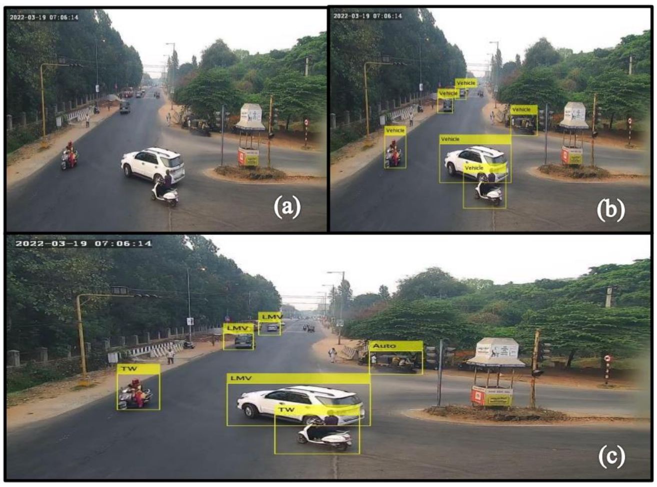


Figure 7. (a) Sample input video-frame; (b) object detection using Yolo-v2 and (c) vehicle type classification from LSTM-OWHO for cloudy condition

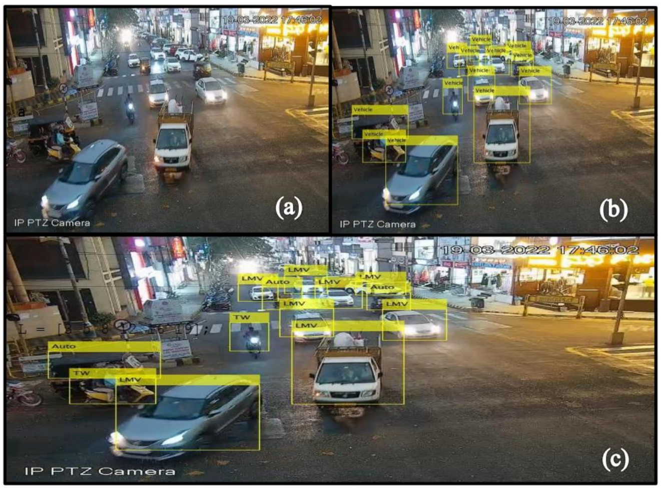
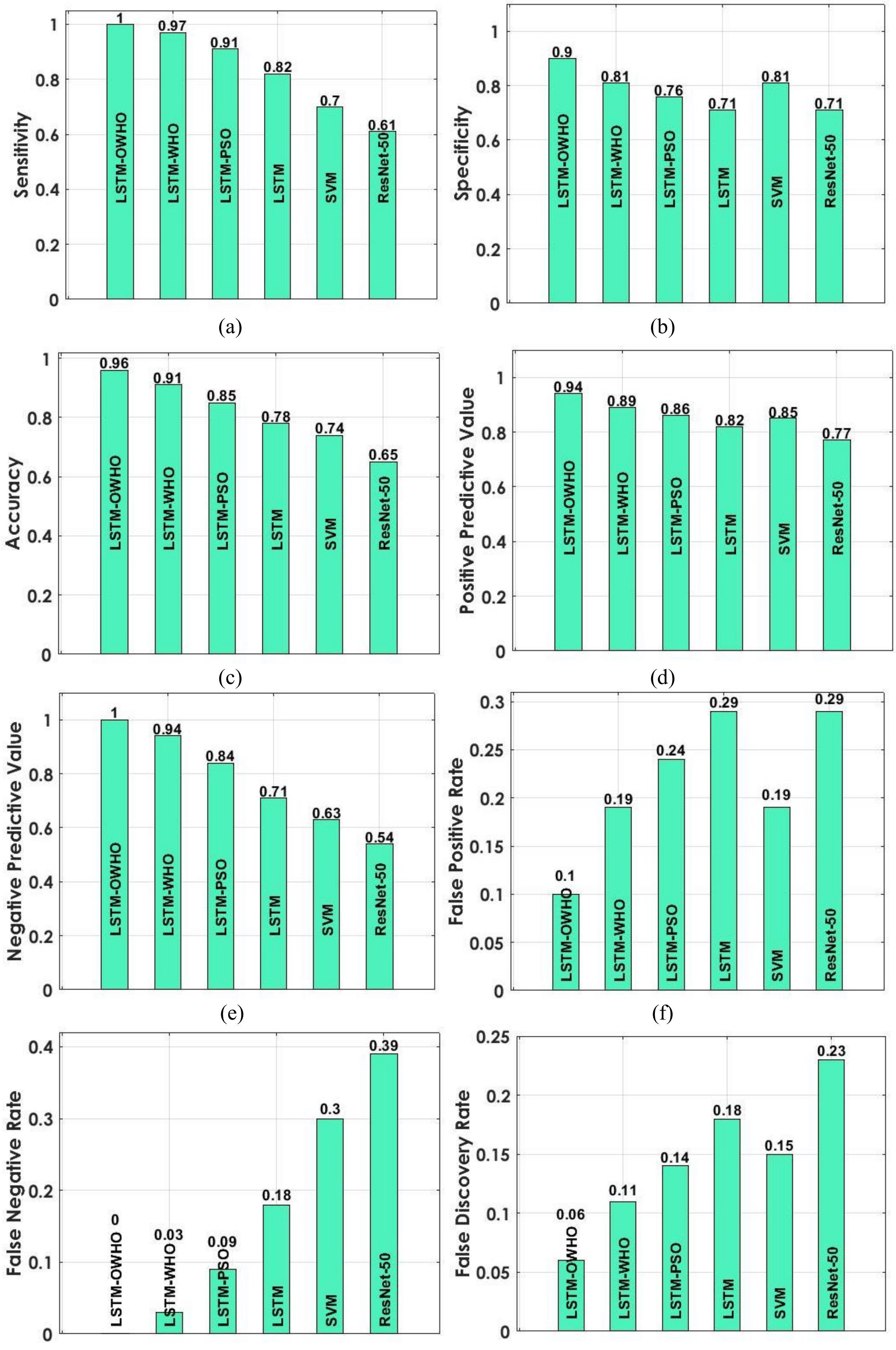


Figure 8. (a) Sample input video-frame; (b) object detection using Yolo-v2 and (c) vehicle type classification from LSTM-OWHO for Night-time condition



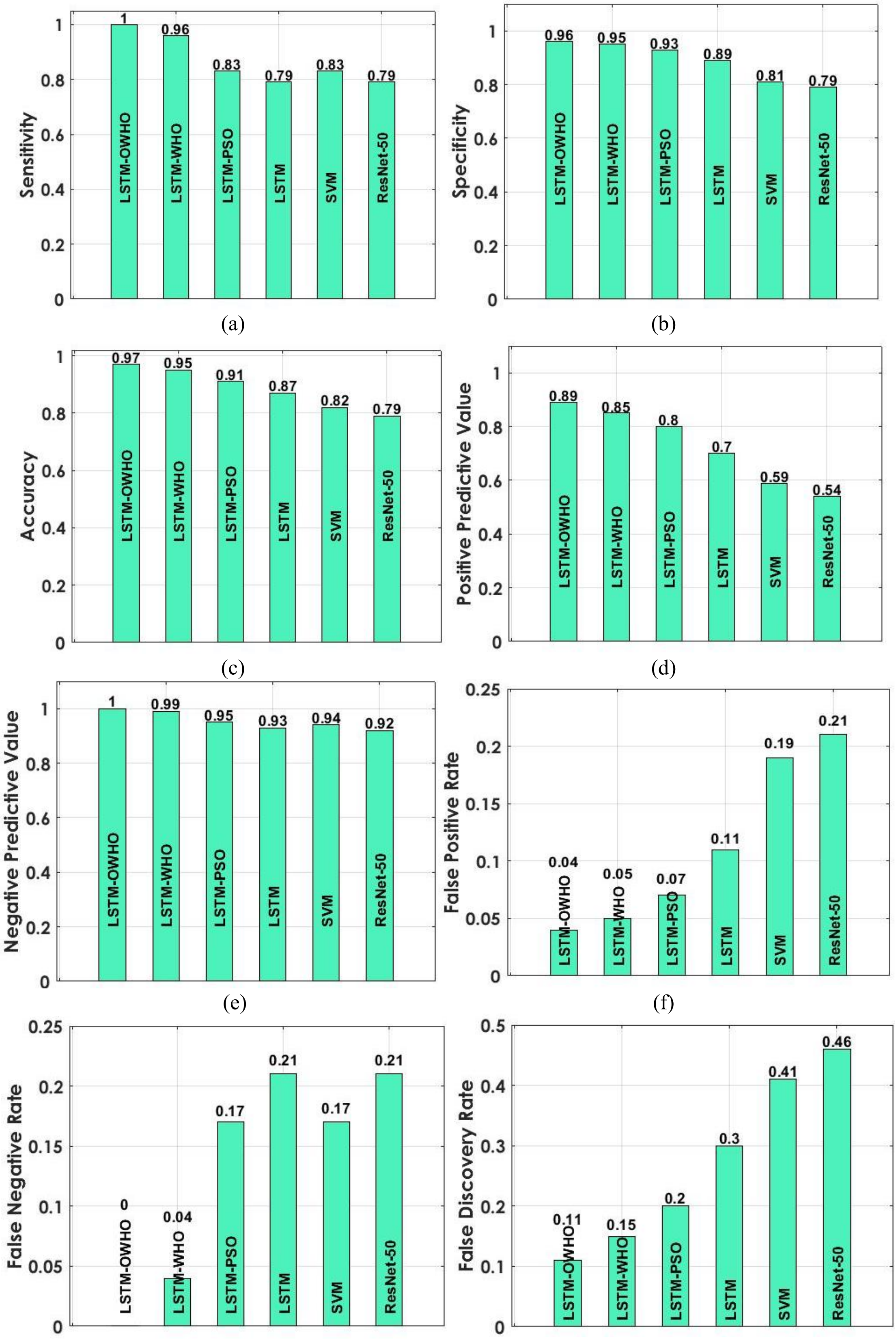
(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 9. Techniques wise vehicle classification performance measure for day-light condition



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 10. Techniques wise vehicle classification performance measure for cloudy condition

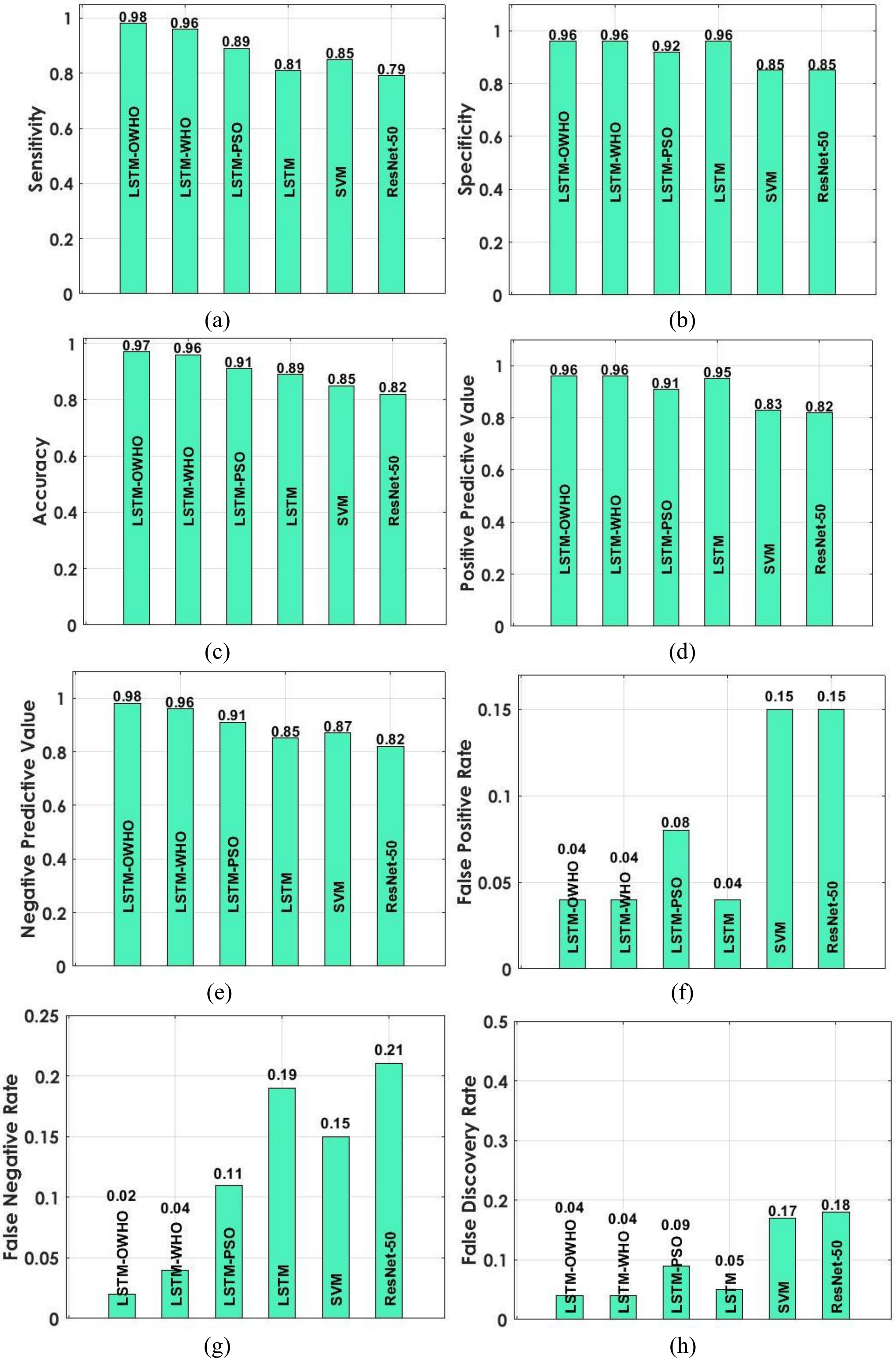


Figure 11. Techniques wise vehicle classification performance measure for Night-light condition

Sensitivity: Sensitivity refers to the test's ability to appropriately detect recognized vehicle that do have expected vehicle characteristics. The mathematical expression (Eq. (11)) for identifying the sensitivity is:

$$Sensitivity = \frac{T_r P}{(T_r P + F_a N)} \qquad (11)$$

The OWHO's association with LSTM achieves 100% in day and cloudy light conditions; during night conditions, the performance is 98%. The performance of proposed approach was superior in comparative techniques and in three different lighting conditions.

Specificity: Specificity relates to the test's ability to appropriately reject not-recognised vehicle without having expected vehicle characteristics. The mathematical expression (Eq. (12)) for identifying the specificity is:

$$Specificity = \frac{T_r N}{(T_r N + F_a P)} \qquad (12)$$

The performance of OWHO-configured LSTM is slightly better than that of the WHO-configured LSTM model and far better than traditional LSTM and other comparative techniques. The outcome of the proposed approach in terms of specificity is 90% for daylight conditions and 96% for both cloudy and night-time lighting conditions.

Accuracy: Accuracy is employed as a statistical measure to effectively identify and distinguish recognized or unrecognized vehicles based on their expected characteristics. The mathematical expression (Eq. (13)) for identifying the accuracy is:

$$Accuracy=\frac{T_rP+T_rN}{(T_rP+T_rN+F_aP+F_aN)} \tag{13}$$

The OWHO-configured LSTM attains 96% accuracy in daylight conditions and 97% accuracy in cloudy and night-time lighting conditions. When the performance is compared with the WHO-configured LSTM model, the accuracy is 91%, 95%, and 96% for daytime, cloudy, and night-time lighting conditions, respectively. When PSO is employed for identifying hyperparameters, the accuracy is 85% for daylight conditions and 91% for cloudy and night-time lighting conditions. The proposed approach has 18% more accuracy in daylight conditions, 10% more accuracy in cloudy conditions, and 8% more accuracy in nightlight conditions. From these results, it is evident that integrating optimization into identifying optimal hyperparameters leads to better performance.

Positive Predictive Value: The OWHO-configured LSTM attains daytime PPV of 94%, in cloudy conditions it is 89%, and in nighttime lighting conditions it is 96%. The mathematical expression (Eq. (14)) for identifying the PPV is:

$$PPV=\frac{T_rP}{(T_rP+F_aP)} \tag{14}$$

Negative Predictive Value: The OWHO-configured LSTM daytime and cloudy conditions PPV of 100%, and in nighttime lighting conditions it is 98%. The mathematical expression (Eq. (15)) for identifying the NPV is:

$$NPV=\frac{T_rN}{(T_rN+F_aN)} \tag{15}$$

False Positive Rate: The FPR is determined by dividing the total number of actually unrecognised events by the number of unrecognised events that were mistakenly classified as recognised vehicle (false positives). The mathematical expression (Eq. (16)) for identifying the FPR is:

$$FPR=\frac{F_aP}{(F_aP+T_rN)} \tag{16}$$

The OWHO-configured LSTM attains daytime FPR of 0.1, and in cloudy and nighttime lighting conditions it is 0.04.

False Negative Rate: The OWHO-configured LSTM attains daytime and cloudy condition FNR of 0 and nighttime lighting conditions it is 0.02. The mathematical expression (Eq. (17)) for identifying the FNR is:

$$FNR=\frac{F_aN}{(F_aN+T_rP)} \tag{17}$$

False Discovery Rate: The FDR is the anticipated ratio of false positive classifications (also known as false discoveries) to all positive classifications. Both the FP and TP counts are included in the overall number of rejections of the null. The OWHO-configured LSTM achieves 0.06 FDR for daytime lighting, 0.11 FDR for cloudy circumstances, and 0.04 FDR for nighttime lighting. The mathematical expression (Eq. (18)) for identifying the FDR is:

$$FDR=\frac{F_aP}{(F_aP+T_rP)} \tag{18}$$

4.1 Confusion matrix

Confusion matrices are employed to visualize important predictive analytics, including sensitivity, specificity, accuracy, and PPV. They offer direct comparisons of values like TP, FP, TN, and FN, making them valuable tools. Each cell in the confusion matrix represents the total number of observations. The rows correspond to the true class, while the columns represent the predicted class. Diagonal cells indicate correctly classified observations, while off-diagonal cells signify misclassifications. Figure 12 explains the performance of the proposed LSTM-OWHO classification approach using the confusion matrix. Table 1 shows the accuracy of dissimilar techniques evaluated under three different lighting conditions, revealing their respective performance.

Table 1. Performance of employed techniques w.r.t accuracy for three different lighting conditions

Techniques	**Lighting-Conditions**		
	Day-Time	**Cloudy Condition**	**Night-Time**
ANN [29]	88.01	-	-
ANFIS [29]	91.13	-	-
ResNet-50	64.81	76.79	82.12
SVM	74.07	81.82	85.51
LSTM	77.78	86.87	89.62
LSTM-PSO	85.19	90.91	91.19
LSTM-WHO	90.74	94.95	96.24
LSTM-OWHO	96.30	96.97	97.36

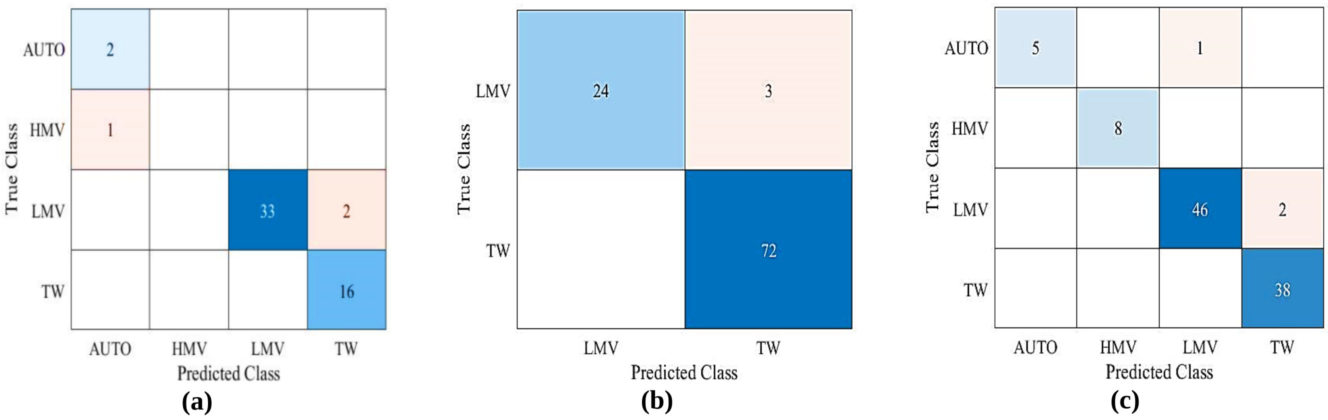


Figure 12. Confusion matrix for (a) Day-time light condition; (b) Cloudy condition; (c) Night time lighting condition

4.2 Limitations of the current approach

Sensitivity to Extreme Environmental Conditions: The current approach may struggle under extreme environmental conditions, like heavy fog or severe storms, impacting its reliability.

Limited Dataset: The research might have used a limited dataset, potentially hindering the model's generalization to diverse vehicle types and scenarios.

Real-Time Processing Constraints: Hardware limitations may affect real-time processing capabilities, leading to delays or computational challenges.

Increased Model Complexity: The combined use of YOLO-v2, ResNet50, and LSTM models may result in heightened computational complexity, especially on resource-constrained devices.

4.3 Opportunities for improvement

Data Augmentation: Enhance the dataset with diverse and challenging scenarios to improve the model's adaptability and robustness.

Transfer Learning: Utilize pre-trained models and fine-tuning to boost accuracy and reduce the need for extensive hyperparameter tuning.

Hardware Optimization: Investigate hardware-specific optimizations to ensure efficient real-time processing.

Advanced Algorithms: Explore newer deep learning architectures or machine learning techniques to potentially enhance real-time vehicle classification.

Robustness Testing: Thoroughly test the model under various real-world conditions to ensure its reliability and accuracy in practical applications.

5. CONCLUSIONS

In this research, a traffic monitoring system was developed and tested under three different lighting conditions, utilizing eight standard performance measures. The system employed ResNet-50 to extract features for use in classification techniques. The results demonstrate that the performance of the techniques improved significantly in classifying vehicle types under all three lighting conditions. OWHO optimization facilitated efficient exploration of the hyperparameter space, initialization of parameters, selection of the right architecture, regularization of the model, adaptation of the learning rate, and optimization of the objective function—all of which collectively contributed to improved LSTM model performance in vehicle classification. By systematically fine-tuning these aspects, OWHO helped the LSTM network achieve higher accuracy and better generalization to real-world data. This research confidently concludes that utilizing OWHO to configure LSTM results in an effective traffic monitoring system with superior performance in terms of convergence speed, accuracy, and stability when compared to alternative techniques. The average accuracy achieved across the three lighting conditions was 97.38%, surpassing that of optimization techniques like LSTM and other comparative classification methods.

The current approach for real-time vehicle classification has limitations concerning sensitivity to extreme environmental conditions, reliance on a potentially limited dataset, constraints in real-time processing, and increased model complexity. Improvement opportunities involve enhancing the dataset through data augmentation, utilizing transfer learning, exploring hardware optimizations, adopting advanced algorithms, refining hyperparameter optimization, and conducting robustness testing in diverse real-world scenarios.

REFERENCES

[1] Li, F., Lee, C.H., Chen, C.H., Khoo, L.P. (2019). Hybrid data-driven vigilance model in traffic control center using eye-tracking data and context data. Advanced Engineering Informatics, 42: 100940. https://doi.org/10.1016/j.aei.2019.100940

[2] Zhao, Z.Q., Zheng, P., Xu, S.T., Wu, X. (2019). Object detection with deep learning: A review. IEEE Transactions on Neural Networks and Learning Systems, 30(11): 3212-3232. https://doi.org/10.48550/arXiv.1807.05511

[3] Khan, S., Ali, H., Ullah, Z., Bulbul, M.F. (2018). An intelligent monitoring system of vehicles on highway traffic. In 2018 12th international conference on open source systems and technologies (ICOSST), pp. 71-75. https://doi.org/10.1109/ICOSST.2018.8632192

[4] Sun, W., Zhang, G., Zhang, X., Zhang, X., Ge, N. (2021). Fine-grained vehicle type classification using lightweight convolutional neural network with feature optimization and joint learning strategy. Multimedia Tools and Applications, 80: 30803-30816. https://doi.org/10.1007/s11042-020-09171-3

[5] Hedeya, M.A., Eid, A.H., Abdel-Kader, R.F. (2020). A super-learner ensemble of deep networks for vehicle-

type classification. IEEE Access, 8: 98266-98280. https://doi.org/10.1109/ACCESS.2020.2997286

[6] Kim, H. (2022). Multiple vehicle tracking and classification system with a convolutional neural network. Journal of Ambient Intelligence and Humanized Computing, 13(3): 1603-1614. https://doi.org/10.1007/s12652-019-01429-5

[7] Feng, X., Jiang, Y., Yang, X., Du, M., Li, X. (2019). Computer vision algorithms and hardware implementations: A survey. Integration, 69: 309-320. https://doi.org/10.1016/j.vlsi.2019.07.005

[8] Elpeltagy, M., Sallam, H. (2021). Automatic prediction of COVID-19 from chest images using modified ResNet50. Multimedia tools and applications, 80(17): 26451-26463. https://doi.org/10.1007/s11042-021-10783-6

[9] Xiao, H., Sotelo, M.A., Ma, Y., Cao, B., Zhou, Y., Xu, Y., Li, Z. (2020). An improved LSTM model for behavior recognition of intelligent vehicles. IEEE Access, 8: 101514-101527. https://doi.org/10.1109/ACCESS.2020.2996203

[10] Nakisa, B., Rastgoo, M. N., Rakotonirainy, A., Maire, F., Chandran, V. (2018). Long short term memory hyperparameter optimization for a neural network based emotion recognition framework. IEEE Access, 6: 49325-49338. https://doi.org/10.1109/ACCESS.2018.2868361

[11] Zheng, R., Hussien, A.G., Jia, H.M., Abualigah, L., Wang, S., Wu, D. (2022). An improved wild horse optimizer for solving optimization problems. Mathematics, 10(8): 1311. https://doi.org/10.3390/math10081311

[12] Naruei, I., Keynia, F. (2022). Wild horse optimizer: A new meta-heuristic algorithm for solving engineering optimization problems. Engineering with Computers, 38(4): 3025-3056. https://doi.org/10.1007/s00366-021-01438-z

[13] Choi, T.J., Togelius, J., Cheong, Y.G. (2021). A fast and efficient stochastic opposition-based learning for differential evolution in numerical optimization. Swarm and Evolutionary Computation, 60: 100768. https://doi.org/10.48550/arXiv.1908.08011

[14] Karungaru, S., Dongyang, L., Terada, K. (2021). Vehicle detection and type classification based on CNN-SVM. International Journal of Machine Learning and Computing, 11(4): 304-310. https://repo.lib.tokushima-u.ac.jp/115186

[15] Han, X., Chang, J., Wang, K. (2021). Real-time object detection based on YOLO-v2 for tiny vehicle object. Procedia Computer Science, 183: 61-72. https://doi.org/10.1016/j.procs.2021.02.031

[16] Hou, X., Zhang, Y., Hou, J. (2020). Application of YOLO V2 in construction vehicle detection. In The International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, pp. 1249-1256 Cham, Springer International Publishing. https://doi.org/10.1007/978-3-030-70665-4_135

[17] Luo, J.Q., Fang, H.S., Shao, F.M., Zhong, Y., Hua, X. (2021). Multi-scale traffic vehicle detection based on faster R–CNN with NAS optimization and feature enrichment. Defence Technology, 17(4): 1542-1554. https://doi.org/10.1016/j.dt.2020.10.006

[18] Şentaş, A., Tashiev, İ., Küçükayvaz, F., Kul, S., Eken, S., Sayar, A., Becerikli, Y. (2020). Performance evaluation of support vector machine and convolutional neural network algorithms in real-time vehicle type and color classification. Evolutionary Intelligence, 13: 83-91. https://doi.org/10.1007/s12065-018-0167-z

[19] Kolukisa, B., Yildirim, V.C., Elmas, B., Ayyildiz, C., Gungor, V.C. (2022). Deep learning approaches for vehicle type classification with 3-D magnetic sensor. Computer Networks, 217: 109326. https://doi.org/10.1016/j.comnet.2022.109326

[20] Rachmadi, R.F., Uchimura, K., Koutaki, G., Ogata, K. (2018). Single image vehicle classification using pseudo long short-term memory classifier. Journal of Visual Communication and Image Representation, 56: 265-274. https://doi.org/10.1016/j.jvcir.2018.09.021

[21] Chen, L., Lin, S., Lu, X., Cao, D., Wu, H., Guo, C., Wang, F.Y. (2021). Deep neural network based vehicle and pedestrian detection for autonomous driving: A survey. IEEE Transactions on Intelligent Transportation Systems, 22(6): 3234-3246. https://doi.org/10.1109/TITS.2020.2993926

[22] AL-Saati, N.A. (2022). The exploration of wild horse optimization in reliability redundancy allocation problems. International Journal of Intelligent Engineering and Systems, 15(4): 198-207.

[23] Mandal, B., Roy, P.K. (2021). Dynamic economic dispatch problem in hybrid wind based power systems using oppositional based chaotic grasshopper optimization algorithm. Journal of Renewable and Sustainable Energy, 13(1): 013306. https://doi.org/10.1063/5.0028591

[24] Arora, N., Kumar, Y. (2022). Automatic vehicle detection system in day and night mode: Challenges, applications and panoramic review. Evolutionary Intelligence, 1-19. https://doi.org/10.1007/s12065-022-00723-0

[25] Lin, C.T., Huang, S.W., Wu, Y.Y., Lai, S.H. (2020). GAN-based day-to-night image style transfer for nighttime vehicle detection. IEEE Transactions on Intelligent Transportation Systems, 22(2): 951-963. https://doi.org/10.1109/TITS.2019.2961679

[26] Bell, A., Mantecon, T., Diaz, C., del-Blanco, C.R., Jaureguizar, F., Garcia, N. (2021). A novel system for nighttime vehicle detection based on foveal classifiers with real-time performance. IEEE Transactions on Intelligent Transportation Systems, 23(6): 5421-5433. https://doi.org/10.1109/TITS.2021.3053863

[27] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. https://doi.org/10.48550/arXiv.1512.03385

[28] Al-Antari, M.A., Han, S.M., Kim, T.S. (2020). Evaluation of deep learning detection and classification towards computer-aided diagnosis of breast lesions in digital X-ray mammograms. Computer Methods and Programs in Biomedicine, 196: 105584. https://doi.org/10.1016/j.cmpb.2020.105584

[29] Murugan, V., Vijaykumar, V.R. (2018). Automatic moving vehicle detection and classification based on artificial neural fuzzy inference system. Wireless Personal Communications, 100: 745-766. https://doi.org/10.1007/s11277-018-5347-8