ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20250117002275
Original scientific paper

Hybrid Black Widow Combined Seagull Optimization with Deep Learning for Efficient Road Classification in UAV-Aided Intelligent Transportation Systems

R. RAJAGANAPATHI, N. PRATHAP, R. HARIHARAN, S. BALAKRISHNAN*

Abstract: Unmanned Aerial Vehicles (UAVs) are pivotal in Intelligent Transportation Systems (ITS) for smart cities, enabling interconnected and autonomous vehicle networks. UAVs enhance ground vehicles by establishing efficient wireless connections and aiding in real-time road monitoring. Artificial Intelligence (AI) and Machine Learning (ML) optimize UAV operations, including dynamic control, path planning, and environmental perception. Deep Learning (DL), particularly Convolutional Neural Networks (CNNs), excels in detection tasks, making it ideal for UAV-based road classification.This study proposes a Hybrid Black Widow Combined Seagull Optimization with Deep Learning-based Classification (HBWCSO-DLC) algorithm to improve road classification accuracy. The HBWCSO-DLC integrates Black Widow Optimization (BWO) and Seagull Optimization (SOA) to enhance feature selection and CNN performance. Evaluated on a road image dataset, HBWCSO-DLC achieves 99.48% accuracy, 98.69% sensitivity, 99.67% specificity, and a 98.69% F1-score, outperforming existing methods like MODAE-RCM, Adam, and SGD. The hybrid optimization ensures robust convergence, while the CNN architecture adapts to complex road textures captured by UAVs.The results demonstrate HBWCSO-DLC's superiority in ITS applications, including autonomous driving and safety systems. This work provides a scalable solution for UAV-assisted road classification, combining bio-inspired optimization with deep learning for real-time, high-precision outcomes.

Keywords: autonomous vehicles; deep learning (DL); intelligent transportation systems (ITS); road classification; unmanned aerial vehicles (UAVs)

1 INTRODUCTION

Intelligent Transportation System (ITS) is a significant position in the intelligent city that can offer quick and comfortable pathways for travelling public and also the transportation of products [1]. Nevertheless, ITS is a data-centric management model and the decision-making of ITS needs a large amount of information to maintain. For instance, vehicle flow, road congestion, and different data will be combined on Roadside Units (RSUs) [2]. Commonly, RSUs are confined by their storage capacity and required to sporadically transmit the information to the information center of ITS. As the arrangement scale of information collected devices like RSUs has become increasingly larger, the protection of ITS becomes increasingly wider and the complication of information collected in information centers has raised [3]. UAVs (Unmanned Aerial Vehicles) act as information mules for collecting information sporadically and it is an efficient information accumulation technique. UAVs have flown unrestrainedly to gather information from every storage device like RSUs [4]. However, the resources of UAVs are also restricted. When the amount of fuel of the UAV is consumed, it requires returning to the information center to refuel. Consequently, UAV flight time is also limited. There are a number of systems to decrease the flight time of UAVs. The majority of the general technique is for centralizing the information of entire nodes (e.g. storage devices) to many cluster heads over the transmission of multi-hopping [5]. Hence, the UAV does not require traveling through entire nodes, which can decrease the flight time and the flight distance of the UAV.

Classification of road surfaces will become a captivating problem [6]. The wide-ranging road categories comprise dry, rough, ice, and curvy road surfaces. In these conditions, many image processing techniques are utilized for supporting intelligent vehicles and for the classification of different categories of the road [7]. Nevertheless, the road image quality is affected by the blurring, illuminated variation, and dynamic weather [8]. Processing large volumes of high-resolution UAV images using traditional Remote Sensing (RS) methods is time-consuming and often unsuitable for time-sensitive applications such as flood management. During the last decades, traditional Machine Learning (ML) classifiers have been utilized for the application of RS mapping. These techniques did not produce important outcomes in the conditions of accuracy owing to the intricate nature of the textual information, particularly within highly urbanized and heavily vegetated zones [9]. Deep Convolutional Neural Networks (CNNs) control these difficulties and have been exposed to produce optimal performance in a number of tasks, comprising detection of objects, identification of objects, image classification, and scene labelling [10]. Dissimilar to other ML classifiers, CNNs offer a hierarchical demonstration of the information by applying different convolutions and can acquire new knowledge features representation from large databases automatically.

2 LITERATURE REVIEW

Silva et al. [11] inspect a novel automated road damage recognition system employing UAV images and DL algorithms. Preserving road structure was critical to making sure a safe and maintainable transportation method. Thus, the author presents utilizing UAVs and AI technology for improving road damage recognition efficacy and accuracy significantly. The presented method employs 3 methods like YOLOv7, YOLOv4, and YOLOv5, for object recognition and localization in UAV images.

In [12], a new crack-affected risk-informed calculation structure was presented for the maintenance and monitoring of roads. According to the height and pixel statistics of identified crack, the real size of cracks is computed employing the pixel-physical conversion co-efficient formula when utilizing a deep learning system for the early detection of road cracks, yellow lane markings, and potholes. For accomplishing the deeper propagation and for preventing saturation under the trained stage, self-regularizing activation was utilized in the first 12 layers with the ReLU activation function. In [14], a light

network structure method (M-YOLO) dependent upon Mobilenet V3-YOLOv5S was presented. SPPNet network design is established for removing repetitive features and enhancing target recognition accuracy.

A road extraction projection was shown in [15-17] using images captured by unmanned aerial vehicles using low altitude remote sensing (LARS). In this method, a U-shaped FCN (or UFCN for short) is an FCN structure that uses a stack of convolutional and mirrored deconvolutions, as well as a way to avoid connecting them so that the local data remains. According to Saetchnikov et al. [18], UAVs can use the S-Y-biLSTM method to shoot objects while they are in motion. Networks based on YOLOv4eff and SSDeff offer mapping extraction features, whereas networks based on BiLSTM offer tracking backbone approaches and object identification systems, respectively. The offered system incorporates these three networks. In addition, the authors suggest a feature fusion element as a residual forecast method for object recognition using the SSD network's full feature layers. To solve the problem of object recognition, this is done.

Unmanned Aerial Vehicles (UAVs) have emerged as a transformative technology in the realm of Intelligent Transportation Systems (ITS), particularly in smart city infrastructure. Their ability to offer a bird's-eye view and real-time data acquisition has enabled substantial advancements in traffic monitoring, vehicle tracking, accident detection, and road condition assessment. Early applications focused primarily on surveillance and traffic flow analysis, leveraging onboard cameras and wireless communication systems. These aerial systems are also used to establish robust Vehicle-to-Everything (V2X) communication networks in areas with limited infrastructure. Moreover, UAVs assist in generating high-resolution maps for navigation and urban planning. Despite their advantages, these systems face challenges related to energy efficiency, real-time processing capabilities, and safe autonomous operations. Nonetheless, the integration of UAVs with intelligent systems has ushered in a new paradigm where data collected from above significantly enhance the functionality and safety of ITS components, laying the groundwork for further innovations in road classification and vehicle autonomy [19].

Artificial Intelligence (AI) and Machine Learning (ML) play pivotal roles in enhancing the autonomy, efficiency, and intelligence of UAVs within ITS applications. These technologies are applied in various UAV tasks such as navigation, object detection, anomaly detection, and path planning. ML algorithms, including Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and Decision Trees, have been widely used for road classification and traffic pattern analysis. Recent studies emphasize the significance of integrating ML models directly onboard UAVs to reduce transmission

In [13], an effort was developed in overhead and facilitate faster processing. Despite the advances, these models require extensive tuning, are data-hungry, and often depend on handcrafted features. These limitations have paved the way for more sophisticated DL-based systems that automatically extract hierarchical features and improve the robustness of UAV-based ITS operations [20].

Deep Learning (DL), particularly Convolutional Neural Networks (CNNs), has revolutionized the field of image classification and pattern recognition, offering high performance in aerial image analysis. CNNs have been extensively employed in road detection and classification tasks due to their powerful feature extraction capabilities. Architectures like AlexNet, VGGNet, ResNet, and U-Net have been applied to aerial imagery captured by UAVs to detect road boundaries, classify pavement conditions, and segment lanes. Despite these advancements, challenges remain, including high computational demands, difficulty in training on heterogeneous datasets, and performance degradation under varying lighting and weather conditions [21]. These limitations suggest a growing interest in hybrid models that combine deep learning with optimization algorithms to refine feature selection and improve classification robustness in real-time UAV applications.

Optimization algorithms have become critical in enhancing the performance of AI and DL models, particularly in high-stakes applications like ITS. Metaheuristic algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), and Whale Optimization Algorithm (WOA) have been employed for feature selection, hyperparameter tuning, and path optimization in UAV operations. These techniques offer global search capabilities and help escape local minima, thereby improving model accuracy and convergence speed. In road classification, hybrid optimization ensures that the deep learning models are trained on the most relevant features, minimizing overfitting and reducing computational complexity. The introduction of newer algorithms like Black Widow Optimization (BWO) and Seagull Optimization Algorithm (SOA) has shown promising results in various engineering applications [22, 23]. Yet, their integration with deep learning models for UAV-based ITS solutions remains underexplored. This presents an opportunity for developing a hybrid optimization–deep learning framework that outperforms conventional models in road classification tasks.

There is an absence of a unified optimization-and-classification approach capable of delivering both high accuracy and computational efficiency for road classification in rapidly changing urban environments. This motivated the development of our proposed HBWCSO-DLC algorithm, which synergistically combines the Black Widow Optimization Algorithm and Seagull Optimization with deep learning to enhance feature selection and classification performance.

3 PROPOSED DESIGN

In this study, we have introduced a novel HBWCSO-DLC system for the road classification process. The Fig. 1 and Fig. 2 signify the workflow and flowchart of the HBWCSO-DLC method respectively.

Feature Extraction: Modified MobileNet Model

Initially, the modified MobileNet model is utilized to generate features of the UAV images. MobileNetVl was established on the basis of the traditional VGG model combined with depth-separable convolution. There have been inverse residual and linear bottleneck characteristics in MobileNetV2. As MobileNetV3 can be presented with

NetAdapt and NAS optimizer, the non-linear function of *h*-swish was exchanged by ReLu, and expensive layers were discarded, which is a revised version of swish non-linearity with better quantization effect and fast computation rate.

The basic concept is to apply the piecewise linear hard analog $\frac{\mathrm{ReLU6}(x+3)}{6}$ for replacing the sigmoid, whereas the nuance takes place in the service of ReLu6 rather than clipping constant. Thus, *h*-swish can be determined as follows:

$$h-\mathrm{swish}[x] = x\mathrm{ReLU6}(x+3)/6 \tag{1}$$

Optimising for ReLU6 is important in the context of any and all hardware and software frameworks. Removed in the quantized mode was the possibility of numerical accuracy loss due to the employment of an approximation sigmoid. The *h*-swish method is run as a piecewise function, which decreases the delay cost by reducing the number of memory accesses.

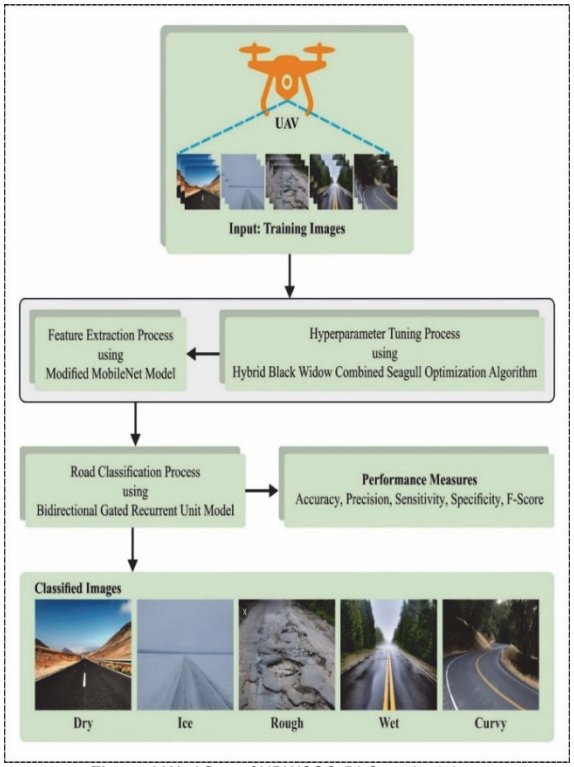

Figure 1 Workflow of HBWCSO-DLC methodology

The SE model is provided by MobileNetV3, which uses the Squeeze function to obtain a 1 × 1 × C vector. In the inverse residual model, this happens after the 3 × 3 depthwise convolution. Production of a 1 × 1 × C vector follows the "Excitation (FC layer-activation)" function. The "FC layer-activation" function's output channel counts have been halved from their original number in an effort to decrease the computation time. Finally, the generated vector was bitwise multiplied by the depthwise convolutional results to change the channel weights and improve the network's performance. Finding and improving the network architecture, which includes the structure and arrangement of blocks, was initially accomplished using the NAS model. Lastly, the channel counts for all filters were determined using the NetAdapt model.

The proposed HBWCSO-DLC framework begins with UAV-based image acquisition, followed by preprocessing to enhance data quality. The CNN is initially trained to extract deep spatial features, and its performance is optimized using a hybrid Black Widow and Seagull Optimization strategy. The BWO ensures effective local exploitation through mating and cannibalism operations, while SOA supports global exploration via migration and attacking behavior. This hybrid mechanism fine-tunes the CNN weights, improving classification accuracy. Finally, the optimized CNN performs road classification, and the system's performance is validated using multiple metrics.

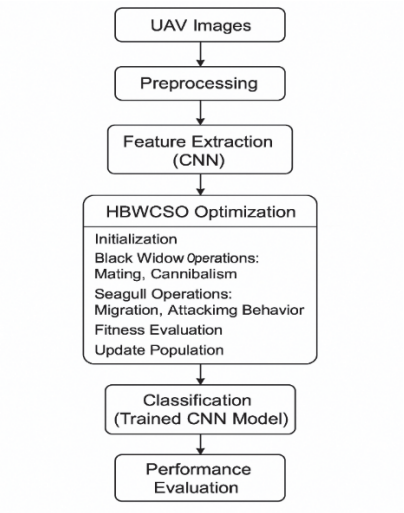

Figure 2 HBWCSO-DLC flow chart

Hyperparameter Tuning: HBWCSO Algorithm

In this work, the hyperparameter tuning of the modified MobileNet model takes place through the design of the HBWCSO approach. During this case, the HBWCSO method was developed by the group of the black widow optimization (BWO) with seagull optimization (SO) algorithm. In the exploration stage of the SO algorithm, a seagull need to meet the requirement: collision avoidance, migration towards the fittest neighbor, and maintaining contact with the optimum search agent. A new search agent location can be evaluated by another A variable to avoid collision between them.

$$\left(X_s\right)^{\circ\rightarrow} = A\cdot\left(P_s\right)^{\circ\rightarrow}(x)$$

where X_s denotes the non-competitive position of the search agent, P_s shows the existing location of the agent, A variable illustrates the movement behaviour of the search agent, and x refers to the existing iteration. The agent switches towards the fittest neighbor after avoiding obstacles between neighbors.

$$\overrightarrow{M_s} = \mathrm{rand}\left((0,1)\cdot\left(\overrightarrow{P_{bs}}(x)\right)-\overrightarrow{P_s}(x)\right) \tag{3}$$

Using what has been learned in the past and during the search process is what exploitation is all about. In the course of an assault, seagulls can alter their speed and

direction at any moment. As the predator strikes its prey, it whips around in a spiral pattern in the air.

The radius of all the spiral turns is represented as *r*. Lastly, $P_s(x)$ stores the fittest solution and moves other agents around. D_s shows the distance among the finest agent and the searching agent.

$$\overrightarrow{P_s}(x) = \left(\overrightarrow{D_s} \cdot x' \cdot y' \cdot z'\right) + \overrightarrow{P_{bs}}(x) \tag{4}$$

This study presents a new hybrid optimization technique, which combines Seagull and Black Widow optimization. Indeed, a hybrid concept in the HBWCSO is capable of enduring the optimization problem in real-time application. Black Widow was conceptualised with the SO algorithm in mind to ensure the greatest possible convergence speed and rate that endure optimal outputs with the target described in this. Presented here is an explanation of the method's mathematical modelling:

Exploration: The purpose of this script is to simulate the flight patterns of a flock of seagulls. All three of these requirements must be met by the seagulls:

Keeping from being hit: The computation of the searching agent's current position uses an additional parameter *A* to avoid collisions between neighbours. Where $\vec{r}_s$ = existing search agent location, $\vec{q}_s$ = search agent location, *u* = existing iteration, *S* = search agent movement behavior.

$$\vec{q}_s = S \cdot \vec{r}_s(u) \tag{5}$$

$$S = v_c - \left(u, \left(v_c | \max_{\text{iteration}}\right)\right) \tag{6}$$

where $u = 0, 1, 2, \ldots, \max_{\text{iteration}}$. The frequency of using parameter *S* can be controlled by v_c which declines linearly from v_c to 0. The level of v_c is fixed as 2.

The optimal neighbor's path of movement: To avoid collisions with its neighbours, the seeking agent moves in the direction of the optimal neighbours.

$$\overrightarrow{T_s} = U \cdot \left(\vec{r}_{ws}(u) - \vec{r}_s(u)\right) \tag{7}$$

where $\vec{r}_s$ = best fit search agent, $\vec{T}_s$ = search agent, and $\vec{r}_{ws}$ = search agent position.

In this work, *B* denotes the arbitrary integer in the cubic chaotic map that can be evaluated as follows:

$$U = 2 \cdot S^2 \cdot VW \tag{8}$$

where, *VW* = arbitrary number.

Keep in touch with the top performing search engine. As a final resort, the seeking agent may adjust its ranking to match the other agents' supremacy, which may be expressed mathematically as follows:

$$\overrightarrow{X_s} = \left|\vec{q}_s + \vec{T}_s\right| \tag{9}$$

The behaviors in the *Y*, and *Z* planes are described in the following.

$$u' = V \cdot \cos(t) \tag{10}$$

$$Y' = V \cdot \sin(t) \tag{11}$$

$$Z' = V \cdot t \tag{12}$$

$$V = uu \cdot ee^{tvv} \tag{13}$$

where *ee* = natural algorithm base, *V* = radius, *T* = random number, and *uu*, *vv* = constant.

Usually, the location of the search agent is updated by Eq. (9) to Eq. (13) that are given below.

$$\vec{r}_s(u) = \left(\vec{X}_s \cdot u' \cdot Y' \cdot Z'\right) + \vec{r}_{ws}(u) \tag{14}$$

According to the proposed logic, BWO location updating can be integrated with seagulls for the best outcomes and subsequently, the updating process is as given in Eq. (15), whereas popl = population.

$$\text{popl} = \text{popl2} + \text{popl3}$$

Furthermore, according to the proposed logic, the cycle crossover can be defined for updating the solution: to generate offspring from their parents where all the slots are filled by the component from dissimilar parents, a phenomenon called Cycle Crossover (CX).

BiGRU Model

Lastly, the BiGRU approach was utilized for road detection and classification procedure. Compared to LSTM, the BiGRU approach is an easy infrastructure and the trained model of it is easier. Initially, the BiGRU exploits gates for suppressing the loss of data. The architecture of the BiGRU technique incorporates forget and the output gates in LSTM for forming the update gate. The GRU module could automatically learn which resource the training can reject and which resource was helpful. The BiGRU architecture has encompassed 2 GRUs that are uni-directional and opposite in direction, integrated into the NN systemprocedure. The two layers of GRU utilize each resource that flows through this network parallel.

The GRU architecture involves resetting and updating gates. The play of reset gate is corresponding to forget and input gates in LSTM that define what amount of data at the prior data is associated with the existing one. The computation equation can be represented as follows:

$$R_t = \sigma\left(W_r \cdot \left[h_{t-1'} X_t\right]\right) \tag{16}$$

Eq. (18) uses R_t to represent the reset gate and W_r to represent the weighted matrix. The input is represented by X_t at time t, while the result was represented by h_{t-1} at time $t-1$. Here, the Hadamard operation was evaluated to define which data is retained and which data should be rejected. The operation could provide results within {0,1}. The update gate defines if upgrade the cell state. Therefore, a sigmoid function was required for mapping the values within zero and one, as follows:

$$Z_t = \sigma\left(W_z \cdot \left[h_{t-1}, X_t\right]\right) \tag{17}$$

The GRU-NN forward propagation equation can be given as follows:

$$\tilde{h}_t = \tan h\left(W^h X_t + U^h\left(h_{t-1} \cdot r_t\right)\right) \tag{18}$$

$$h_t = \left(1 - z_t\right) \cdot \tilde{h}_t + z_t \cdot h_{t-1} \tag{19}$$

From the expression, W^h and U^h show the weighted matrix of GRU, $\tilde{h}_t$ denotes the candidate hidden layer, h_t denotes the hidden layer, and · refers to the multiplication component.

A BiGRU design combines two unidirectional GRUs. In order to get the most accurate result, input is constantly being sent to two GRUs that are working in opposing directions at the same time, and the output is jointly specified.

The output of BiGRU is defined below

$$H = \vec{h}_t \oplus \overleftarrow{h}_t \tag{20}$$

In Eq. (20), $\vec{h}_t$ and $\overleftarrow{h}_t$ show two unidirectional GRUs, H denotes the output of BiGRU $\oplus$ and represents the addition element.

Mathematical Formulation of HBWCSO-DLC Algorithm

1. Fitness Function

The fitness function is used to evaluate the performance of the CNN in road classification. Typically, it is based on classification accuracy or loss.

Accuracy-based fitness function:

$$\text{Fitness} = 1 - \text{Accuracy} = (1/N) \cdot \Sigma(1(\hat{y}_i \neq y_i))$$

Cross-Entropy Loss-based fitness function:

$$\text{Fitness} = -(1/N) \cdot \Sigma(y_i \cdot \log(\hat{y}_i))$$

2. Black Widow Optimization (BWO)

Mating (Crossover) Operation:

$$\text{Child} = \alpha \cdot \text{Parent}_1 + (1 - \alpha) \cdot \text{Parent}_2$$

Mutation (Optional):

$$x_{i_{\text{new}}} = x_i + \delta \cdot \text{randn}()$$

Cannibalism: Top-performing solutions are retained, weakest are eliminated.

3. Seagull Optimization Algorithm (SOA)

Migration Behavior:

$$X^{t+1} = X^t + A * \sin(B \cdot \theta)$$

Attacking Behavior (Spiral Motion):

$$X^{t+1} = X^t \cdot e^{(-\lambda t)} \cdot \cos(2\pi t)$$

Where: α: Crossover rate in BWO; δ: Mutation scaling factor; A, B, λ: Seagull optimization parameters; θ: Spiral angle for migration

4 RESULTS AND DISCUSSION

Five categories (Dry-2500, Ice-2500, Rough-2500, Wet-2500, Curvy-2500) into which the dataset consisting of 12500 samples is used to assess the results of the road classification method created by HBWCSO-DLC. You may see some examples of the photos in Fig. 2.

Dataset Details

The dataset used for training and testing the HBWCSO-DLC algorithm is a labeled road classification dataset containing diverse road images captured by UAVs. It consists of approximately 3500 images categorized into different types of roads including urban roads, highways, rural paths, and curved roads. The images were resized to 224 × 224 pixels for input into the CNN architecture.

Data Preprocessing

Preprocessing steps included image normalization to scale pixel values between 0 and 1, and data augmentation techniques such as horizontal flipping, rotation, and zooming to improve generalization.

Training Configuration

The HBWCSO-DLC algorithm was trained using a custom CNN model optimized through a hybrid metaheuristic approach combining Black Widow Optimization and Seagull Optimization. The training setup was as follows:

- Training Epochs: 50
- Batch Size: 32
- Optimizer: Adam (initial configuration)
- Learning Rate: 0.001
- Hardware: NVIDIA RTX 3080 GPU, 32 GB RAM, Ubuntu 20.04 OS
- Total Training Time: Approximately 2.3 hours

Figure 2 Sample images

In Fig. 3, the road classification results of the HBWCSO-DLC methodology are examined in the procedure of confusion matrix. The outcome indicates that the HBWCSO-DLC technique has properly detected five class labels.

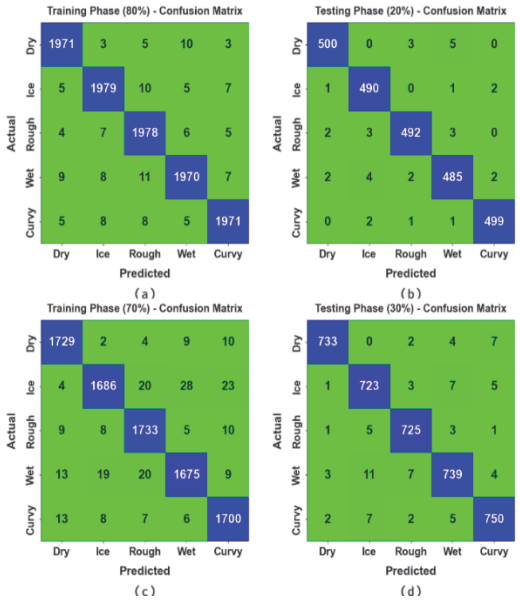

Figure 3 Confusion matrices of (a-b) 80:20 of TR set/TS set and (c-d) 70:30 of TR set/TS set

At 80:20 of the TR set/TS set, the HBWCSO-DLC algorithm's classifier output is examined in Tab. 1.

Table 1 Classifier outcome of HBWCSO-DLC system on 80:20 of TR set/TS set

Class	$Accu_y$	$Prec_n$	$Sens_y$	$Spec_y$	F_{score}
Training Phase (80%)					
Dry	99.56	98.85	98.95	99.71	98.90
Ice	99.47	98.70	98.65	99.67	98.68
Rough	99.44	98.31	98.90	99.58	98.60
Wet	99.39	98.70	98.25	99.67	98.48
Curvy	99.52	98.90	98.70	99.73	98.80
Average	99.48	98.69	98.69	99.67	98.69
Testing Phase (20%)					
Dry	99.48	99.01	98.43	99.75	98.72
Ice	99.48	98.20	99.19	99.55	98.69
Rough	99.44	98.80	98.40	99.70	98.60
Wet	99.20	97.98	97.98	99.50	97.98
Curvy	99.68	99.20	99.20	99.80	99.20
Average	99.46	98.64	98.64	99.66	98.64

The HBWCSO-DLC system investigation revealed a 70:30 ratio between the TR and TS sets, as illustrated in Tab. 2.

Table 2 Classifier outcome of HBWCSO-DLC algorithm on 70:30 of TR set/TS set

Class	$Accu_y$	$Prec_n$	$Sens_y$	$Spec_y$	F_{score}
Training Phase (70%)					
Dry	99.27	97.79	98.57	99.44	98.18
Ice	98.72	97.85	95.74	99.47	96.79
Rough	99.05	97.14	98.19	99.27	97.66
Wet	98.75	97.21	96.49	99.32	96.85
Curvy	99.02	97.03	98.04	99.26	97.53
Average	98.96	97.41	97.41	99.35	97.40
Testing Phase (30%)					
Dry	99.47	99.05	98.26	99.77	98.65
Ice	98.96	96.92	97.83	99.24	97.37
Rough	99.36	98.11	98.64	99.54	98.37
Wet	98.83	97.49	96.73	99.36	97.11
Curvy	99.12	97.78	97.91	99.43	97.85
Average	99.15	97.87	97.87	99.47	97.87

The HBWCSO-DLC technique's TR_{accu_y} and VL_{accu_y} training accuracy values are shown in Fig. 4 and Fig. 5, with the TR set and TS set divided 70:30. While VL_{accu_y} is calculated by evaluating performance on a separate testing database, TL_{accu_y} is determined by estimating the HBWCSO-DLC method on TR datasets.

Figure 4 $Accu_y$ curve of HBWCSO-DLC approach on 70:30 of TR set/TS set

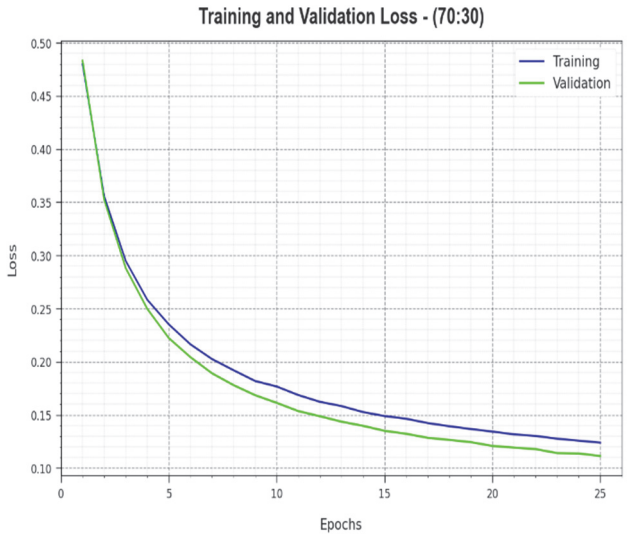

Figure 5 Loss curve of HBWCSO-DLC approach on 70:30 of TR set/TS set

Table 3 Comparative outcome of HBWCSO-DLC approach with other methodologies

Methods	$Accu_y$	$Sens_y$	$Spec_y$	$F1_{score}$
HBWCSO-DLC	99.48	98.69	99.67	98.69
MODAE-RCM	99.07	98.05	99.06	98.14
RMSProp	97.65	97.51	99.05	97.28
Adam	98.62	98.09	98.57	97.89
Adamax	95.39	96.27	98.59	95.87
Adagrad	79.99	80.51	95.07	80.86
SGD	74.11	73.46	93.46	71.13
Adadelta	71.37	71.39	92.51	71.51

In Tab. 3 and Fig. 6, the CT effects of the HBWCSO-DLC algorithm are to compare with the recent systems. The outcomes exhibit the poorer performance of the Adagrad, SGD, and RMSProp methods. Simultaneously, the Adadelta, Adam, and Adamax

optimizers show relatively better outcomes over other techniques.

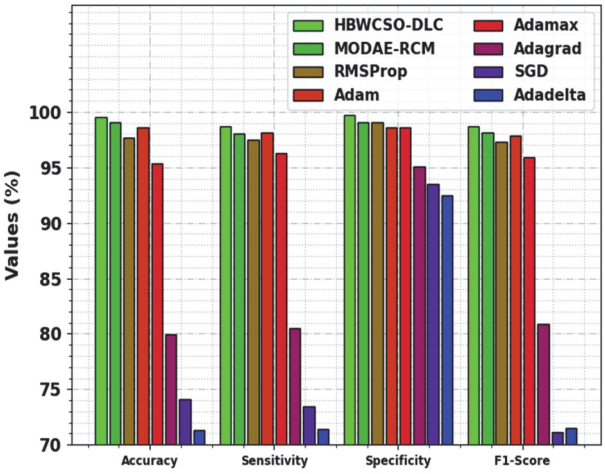

Figure 6 Comparative outcome of HBWCSO-DLC system with other methodologies

Table 4 CT outcome of HBWCSO-DLC approach with other methodologies

Methods	Computational Time / sec
HBWCSO-DLC	08.17
MODAE-RCM	10.09
RMSProp	23.20
Adam	25.10
Adamax	15.90
Adagrad	25.00
SGD	23.40
Adadelta	20.30

Afterwards, the MODAE-RCM technique demonstrated the near-optimal outcomes, the HBWCSO-DLC approach proved a greater performance with a smaller CT of 8.17 s. These performances confirmed the enhancements of the HBWCSO-DLC method through other methods.

5 CONCLUSION

Within the framework of this inquiry, we have presented a unique HBWCSO-DLC strategy for the procedure of road classification. In the Intelligent Transportation System (ITS) that is enabled by unmanned aerial vehicles (UAVs), the classification of the several types of roads that are included is made feasible by the HBWCSO-DLC technique. The HBWCSO-DLC technique that has been supplied makes major use of a modified version of the MobileNet model for the purpose of feature extraction. This change was made in order to simplify the process. The HBWCSO-DLC system that is being suggested now incorporates three basic techniques: a modified MobileNet feature extraction, HBWCSO-based hyper-parameter tuning, and BiGRU-based road classification. These are the three primary methods that are incorporated in the system. Additionally, the hyper-parameter adjustments of the improved MobileNet model may be applied through the architecture of the HBWCSO system. This modification may be implemented. When the simulation results of the HBWCSO-DLC approach are assessed, the road image database is utilized. The outcome may be evaluated using a wide range of various metrics, depending on the specifics of the situation. The HBWCSO-DLC technique provided the best degree of efficiency when compared to other algorithms, as shown by the comprehensive simulation results. This was proved by the effectiveness of the strategy. In the future the ensemble voting technique will be established with the intention of enhancing the performance of the HBWCSO-DLC algorithm.

6 REFERENCES

[1] Outay, F., Mengash, H. A., & Adnan, M. (2020). Applications of unmanned aerial vehicle (UAV) in road safety, traffic and highway infrastructure management: Recent advances and challenges. *Transportation Research Part A: Policy and Practice*, *141*. https://doi.org/10.1016/j.tra.2020.09.018

[2] Jenie, Y. I., Van Kampen, E. J., Ellerbroek, J., & Hoekstra, J. M. (2016). Taxonomy of conflict detection and resolution approaches for unmanned aerial vehicle in an integrated airspace. *IEEE Transactions on Intelligent Transportation Systems*, *18*(3). https://doi.org/10.1109/TITS.2016.2580219

[3] Dai, M., Su, Z., Xu, Q., & Zhang, N. (2021). Vehicle assisted computing offloading for unmanned aerial vehicles in smart city. *IEEE Transactions on Intelligent Transportation Systems*, *22*(3). https://doi.org/10.1109/TITS.2021.3052979

[4] Kataev, M. Y., Kartashov, E. Y., & Avdeenko, V. D. (2023). Method for detecting road defects using images obtained from unmanned aerial vehicles. *Computer Optics*, *47*(3). https://doi.org/10.18287/2412-6179-CO-1209

[5] Han, D., Lee, S. B., Song, M., & Cho, J. S. (2021). Change detection in unmanned aerial vehicle images for progress monitoring of road construction. *Buildings*, *11*(4). https://doi.org/10.3390/buildings11040150

[6] Astor, Y., Utami, R., Winata, S. N., Rahman, F. A., Gustaman, F. A., & Firdaus, M. R. (2022). 3D Model of Pavement Distress Based on Road Gradient Using Unmanned Aerial Vehicle. *Conference on Broad Exposure to Science and Technology 2021, BEST 2021*. https://doi.org/10.2991/aer.k.220131.013

[7] Jia Yi, T. & Ahmad, A.B. (2023). Quality Assessments of Unmanned Aerial Vehicle (UAV) and Terrestrial Laser Scanning (TLS) Methods in Road Cracks Mapping. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *48*. https://doi.org/10.5194/isprs-archives-XLVIII-4-W6-2022-183-2023

[8] Zhou, H., Kong, H., Wei, L., Creighton, D., & Nahavandi, S. (2014). Efficient road detection and tracking for unmanned aerial vehicle. *IEEE Transactions on Intelligent Transportation Systems*, *16*(1). https://doi.org/10.1109/TITS.2014.2331353

[9] Utami, R., Astor, Y., Gustaman, F. A., Firdaus, M. R., Rahman, F. A., & Winata, S. N. (2023). Survey of road distress during the COVID-19 pandemic using unmanned aerial vehicles (UAVs). *AIP Conference Proceedings*, *2646*(1). https://doi.org/10.1063/5.0112707

[10] Kim, N. & Bodunkov, N. (2018). Automated decision making in road traffic monitoring by on-board unmanned aerial vehicle system. *Computer Vision in Control Systems-3: Aerial and Satellite Image Processing*. https://doi.org/10.1007/978-3-319-67516-9_6

[11] Silva, L. A., Leithardt, V. R. Q., Batista, V. F. L., González, G. V., & Santana, J. F. D. P. (2023). Automated Road Damage Detection using UAV Images and Deep Learning Techniques. *IEEE Access*. https://doi.org/10.1109/ACCESS.2023.3287770

[12] Hong, Z., Yang, F., Pan, H., Zhou, R., Zhang, Y., Han, Y., Wang, J., Yang, S., Chen, P., Tong, X., & Liu, J. (2021). Highway crack segmentation from unmanned aerial vehicle

images using deep learning. *IEEE Geoscience and Remote Sensing Letters*, *19*. https://doi.org/10.1109/LGRS.2021.3129607
[13] NasiruddinKhilji, T., Lopes AmaralLoures, L., & Rezazadeh Azar, E. (2021). Distress recognition in unpaved roads using unmanned aerial systems and deep learning segmentation. *Journal of Computing in Civil Engineering*, *35*(2). https://doi.org/10.1061/(ASCE)CP.1943-5487.0000933
[14] Ji, A., Xue, X., Wang, Y., Luo, X., & Wang, L. (2021). Image-based road crack risk-informed assessment using a convolutional neural network and an unmanned aerial vehicle. *Structural Control and Health Monitoring*, *28*(7). https://doi.org/10.1002/stc.2749
[15] Hassan, S. A., Rahim, T., & Shin, S. Y. (2021). An improved deep convolutional neural network-based autonomous road inspection scheme using unmanned aerial vehicles. *Electronics*, *10*(22). https://doi.org/10.3390/electronics10222764
[16] Liu, Y., Shi, G., Li, Y., & Zhao, Z. (2022). M-YOLO based detection and recognition of highway surface oil filling with unmanned aerial vehicle. *2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP)*. https://doi.org/10.1109/ICSP54964.2022.9778782
[17] Kestur, R., Farooq, S., Abdal, R., Mehraj, E., Narasipura, O., & Mudigere, M. (2018). UFCN: A fully convolutional neural network for road extraction in RGB imagery acquired by remote sensing from an unmanned aerial vehicle. *Journal of Applied Remote Sensing*, *12*(1). https://doi.org/10.1117/1.JRS.12.016020
[18] Saetchnikov, I., Skakun, V., & Tcherniavskaia, E. (2022). Robust SY-biLSTM object tracking method for on-road objects shoot from an unmanned aerial vehicle. *2022 IEEE 9th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*. https://doi.org/10.1109/MetroAeroSpace54187.2022.9855900
[19] Wang, L., Shi, W., Tang, Y., Liu, Z., He, X., Xiao, H., & Yang, Y. (2023). Transfer Learning-Based Lightweight SSD Model for Detection of Pests in Citrus. *Agronomy*, *13*(7). https://doi.org/10.3390/agronomy13071710
[20] Gao, Z., Li, Z., Luo, J., & Li, X. (2022). Short text aspect-based sentiment analysis based on CNN + BiGRU. Applied Sciences, 12(5). https://doi.org/10.3390/app12052707
[21] Ahmed Hamza, M., Alqahtani, H., Elkamchouchi, D. H., Alshahrani, H., Alzahrani, J. S., Maray, M., Ahmed Elfaki, M., & Aziz, A. S. A. (2022). Hyperparameter Tuned Deep Autoencoder Model for Road Classification Model in Intelligent Transportation Systems. *Applied Sciences*, *12*(20). https://doi.org/10.3390/app122010605
[22] Mohammadzadeh, A., Javaheri, D., & Artin, J. (2023). Chaotic hybrid multi-objective optimization algorithm for scientific workflow scheduling in multisite clouds. *Journal of the Operational Research Society*, *75*(2), 314-335. https://doi.org/10.1080/01605682.2023.2195426
[23] Verma, G. (2022). Hybrid Optimization Model for Secure Task Scheduling in Cloud: Combining Seagull and Black Widow Optimization. *Cybernetics and Systems*, *55*(8), 2489-2511 https://doi.org/10.1080/01969722.2022.2157609

Contact information:

R. RAJAGANAPATHI
Department of ECE,
Anjalai Ammal Mahalingam Engineering College, Tamil Nadu 614403, India
E-mail: rajume1974@gmail.com

N. PRATHAP
Department of ECE,
Anjalai Ammal Mahalingam Engineering College, Tamil Nadu 614403, India
E-mail: prathap27n@gmail.com

R. HARIHARAN
Department of ECE,
University College of Engineering Thirukkuvalai, Tamil Nadu 610204, India
E-mail: raamhari@gmail.com

S. BALAKRISHANAN
(Corresponding Author)
Department of ECE,
Sona College of Technology (Autonomous), Salem. India
E-mail: balakrishnan.srini@gmail.com