Journal of Theoretical and Applied Information Technology
31st May 2024. Vol.102. No. 10
© Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

INTELLIGENT VEHICLE DETECTION AND CLASSIFICATION IN AERIAL IMAGERY: LEVERAGING ARFOA-ENHANCED LINEAR DISCRIMINANT ANALYSIS WITH ADVANCED VEHICLE PROPOSAL NETWORK

KRANTHI KUMAR LELLA[1], HANUMANTHA RAO JALLA[2], SANDHYA K[3], TEJESH REDDY SINGASANI[4], MOUNIKA B[5], SHUBHANGI JOTEPPA[6], SRINIVASA RAO VEMULA[7], RAMESH VATAMBETI[8*]

[1] School of Computer Science and Engineering, VIT-AP University, Vijayawada 522237, India.
1kranthi1231@gmail.com,
[2] Department of Computer Science, Tikkavarapu Rami Reddy Government Degree College, Kandukur
523105, India. hanucs2000@gmail.com,
3 School of Computing, Mohan Babu University, Tirupati 517102, India. 3sandhyachurchil@gmail.com,
[4] School of Computer and Information Sciences, University of Cumberlands, Louisville, KY 40769, USA.
[5] Department of Information Technology, SRKR Engineering College, Bhimavaram 534204, India.
5bmounika@srkrec.ac.in
[6] Department of Information Technology, MLR Institute of Technology, Hyderabad 500043, India.
6shubhangijoteppa@gmail.com,
[7] Software Test Analyst Senior, FIS Management Services, Durham, North Carolina 27703-8589, USA.
7srinivas.vemula@fisglobal.com
[8] School of Computer Science and Engineering, VIT-AP University, Vijayawada 522237, India.
8v2ramesh634@gmail.com

ABSTRACT

In the realm of computer vision, the detection of vehicles in aerial photography holds significant importance for various applications. Traditional methods rely on computationally intensive techniques with limited effectiveness in handling small objects like vehicles in large-scale aerial images. Recent advancements in deep learning, particularly R-CNNs, have shown promise but are hindered by challenges such as small object detection and the high cost of human annotation for training data. In response, this research proposes a novel system for efficient and accurate vehicle detection. Our approach utilizes a combination of deep learning techniques, including an encoder-decoder architecture for image segmentation and a hyper feature map for precise vehicle proposal generation. Additionally, we introduce the VCLDA model for vehicle classification, fine-tuned using the ARFOA algorithm. Experimental results demonstrate significant performance improvements, achieving detection rates of 84% on the Vehicle Aerial Imagery dataset, 73% on the Vehicle Finding in Aerial Imagery (VEDAI) dataset, and 64% on the German Aerospace Centre (DLR) DLR3K datasets. The proposed system has diverse potential applications, including traffic monitoring, congestion detection, intersection analysis, vehicle categorization, and pedestrian safety measures.

Keywords: *Accurate-Vehicle-Proposal-Network; Artificial Root Foraging Optimizer Algorithm; Region-Based Convolutional Neural Networks; Vehicle Detection; Vehicle Classification Based Linear Discriminant Analysis.*

1. INTRODUCTION

Many people are interested in disaster relief, security, traffic flow monitoring, military target reconnaissance, and vehicle remote sensing due to the important role vehicles play in these detections [1]. The majority of current vehicle detection research is centered around visual pictures. Nighttime reconnaissance of military vehicles or monitoring traffic flow in foggy weather are just two examples of the various detection tasks that require low-light or severe weather conditions [2-3]. However, under low light and poor visibility, visible sensors will not function properly. Research on vehicle recognition in aerial photos is inspired by the fact that infrared sensors can operate continuously even in bad weather [4].

Many people have been paying attention to the problem of vehicle detection in aerial photographs recently because of its importance for many different applications [5]. The small size (as small as 30 × 12 picture elements), different varieties, and changeable orientation of cars make vehicle recognition a tough challenge [6]. False positives can also occur when there are many structures (such as road markings or air conditioning units on buildings) that resemble automobiles [7]. Another factor that makes vehicle recognition more challenging is the restricted processing time for real-time applications. Prior research has suggested several methods for vehicle recognition in aerial photos [8]. The standard practice involves a sliding-window search, which entails scanning each image from every angle and at varying sizes. Support vector machine (SVM) classifiers, AdaBoost classifiers, or features based on shallow learning are employed to check if a vehicle is present in each window [9, 10]. Some approaches rely on road databases as prior knowledge to identify cars on roads, which are unfit for generic situations [11].

The use of deep Convolutional Neural Networks [12], Fast Region-based R-CNN [13], Single Shot Multibox Detector (SSD) [14], and recognition of sensing images based on CNN has been explored. All of these approaches rely on regular rectangles to frame and identify targets [14]. However, there are many scenarios where accomplishing semantic segmentation would be ideal—that is, correctly discerning the shape of each structure, road, river, car, etc.—by simply using the target's shape as a locating and segmenting cue [15]. One area where AI researchers have been focusing a lot of attention recently is semantic segmentation. By deciphering the meaning of a picture based on its pixels' locations and values, it may transform raw data (such as a flat image) into a mask with highlighting [16].

An intriguing topic for traffic monitoring systems that employ aerial video from drones and closed-circuit television cameras is vehicle detection, and this paper focuses on that. In order to effectively control traffic, our study has suggested a new approach that involves picture segmentation, vehicle detection, and classification. Semantic segmentation is initially performed on aerial photos. The next step is to use an AVPN to find cars in the segmented picture. Next, the identified cars are sorted into seven groups using an ARFOA model based on LDA. Additionally, experiments conducted over the VAID, VEDAI, and DLR3K datasets at the German Aerospace Centre verify the provided model. When compared to other state-of-the-art (SOTA) tactics, the experimental results showed that ours had far higher detection and classification accuracy.

1.1 Problem Statement

Detection and classification of vehicles in aerial photography present significant challenges due to the limitations of traditional methods in handling small objects like vehicles in large-scale images. Moreover, existing deep learning approaches face obstacles such as small object detection and the high cost of human annotation for training data. These limitations hinder the development of efficient and accurate vehicle detection systems for applications such as traffic monitoring, congestion detection, and intersection analysis.

1.2 Research Objectives

- To develop a novel system for efficient and accurate vehicle detection in aerial imagery by leveraging deep learning techniques.
- To address the challenges of small object detection and high annotation costs by proposing a combination of encoder-decoder architecture for image segmentation and a hyper feature map for precise vehicle proposal generation.
- To introduce the VCLDA model for vehicle classification, fine-tuned using the ARFOA algorithm, to improve detection accuracy.
- To evaluate the performance of the proposed system on benchmark datasets, including the Vehicle Aerial Imagery dataset, the Vehicle Finding in Aerial Imagery (VEDAI) dataset, and the German Aerospace Centre (DLR) DLR3K datasets.

Here is how the rest of the paper is arranged: Section 2 delivers an overview of relevant literature, Section 3 details the suggested tactic, Section 4 delves into the analysis of the data, and Section 5 draws conclusions.

2. RELATED WORK

A process for creating synthetic datasets using Blender software and aerial photography has been developed by Orić et al. [17]. The pipeline for creating the dataset consists of seven phases, which yield the desired number of photos boxes in the COCO and YOLO formats. This pipeline's steps were followed to create the synthetic dataset, comprising five thousand 2048 x 2048 photos from various locations worldwide with automobiles added onto the roads and highways. We believe that this dataset, along with the associated pipeline, may be very significant for vehicle identification, facilitating the tailoring of models to specific situations and requirements.

To address the issue of the lack of such a dataset, Mustafa & Alizadeh [18] presented a dataset of 2,160 photographs of automobiles on roads in the Region. The Air 2 drone captured the photos in the proposed collection in the Iraqi cities of Erbil and Sulaymaniyah. The images are classified into five categories: personal automobile, truck, bus, taxi, and motorbike. Data collection considered various factors, including different vehicle sizes, weather conditions, illumination, and large camera motions. The photos in our suggested dataset underwent pre-processing and data augmentation techniques, such as auto-orientation and brightness adjustment, which can be utilized to create effective deep learning (DL) models. Following the use of these augmentation approaches, the number of photos was increased to 5,353 for vehicles, 1,500 for taxis, 1,192 for trucks, and 282 and 176 for the other classes.

The cross-modal aerial remote sensing image object detection (CRSIOD) network, proposed by Wang et al. [19], efficiently learns various target characteristics and circumstances. To guide the object detection network as it performs several feature processing tasks, we first construct an illumination perception module. Secondly, we incorporate modality measurements and use them as weights to encourage the network to train in a way that maximizes object detection while minimizing the drawbacks of each modality. Furthermore, we use the cross-modality attentive feature fusion (CMAFF) module to fully extract complementary network features to improve the learning of each of the three modal features independently, and build a two-stream backbone network based on the attention mechanism to improve the learning of challenging samples in the object detection network. Lastly, we upgrade the horizontal detection head to a revolving one to maintain object orientation to optimize detection results. We tested the suggested technique CRSIOD using the public UAV aerial picture dataset from Drone Vehicle. CRSIOD achieves state-of-the-art detection performance when compared to currently used approaches.

The Intelligent Water Drop approach proposed by Vaiyapuri et al. [20] is intended to be used with remote sensing applications. The IWDADL-VDC method utilizes a DL model that has been hyperparameter-tuned for vehicle detection and classification. The two main steps of the IWDADL-VDC approach are vehicle detection and classification, which are achieved through enhanced YOLO-v7 model for vehicle detection and Deep Long Short-Term Memory (DLSTM) technique for categorization. This work utilized the IWDA-based hyperparameter tuning procedure to improve the classification results of the DLSTM model. Experimental validation using a benchmark dataset showed promising results for the IWDADL-VDC technique compared to other recent methods.

To address vehicle detection in UAVs, Sun et al. [21] proposed a new dataset called EVD4UAV, consisting of 90,886 fine-grained tagged cars and 6,284 photos. The dataset is altitude-sensitive and includes several elevations (50, 70, and 90 meters), vehicle characteristics (color, type), and bounding boxes, along with views of visible vehicle roofs. The EVD4UAV dataset was targeted by three traditional deep neural network-based object detectors using white-box and attack techniques. Experimental findings demonstrated that these typical assault strategies were unable to carry out reliable, altitude-insensitive attacks.

Aero-YOLO is a lightweight recognition technique based on YOLOv8 proposed by Shao et al. [22]. The particular method aims to decrease model parameters, increase computational efficiency, and expand the receptive field by replacing the C2f module with C3 and the original. Additionally, the CoordAtt and shuffle care techniques improve feature extraction, which is beneficial for identifying tiny vehicles from a UAV perspective. In conclusion, three novel parameters are suggested to fulfill the demands of various application contexts. Experimental assessments using the VisDrone2019 and UAV-ROD datasets showed that the algorithm suggested in this work enhances the speed and accuracy of identifying vehicles and pedestrians and performs well in a variety of heights, angles, and imaging situations.

Using the VisDrone-DET dataset, Muzammul et al. [23] presented a novel technique for aerial image analysis by fusing the Slicing Aided Hyper Inference (SAHI) methodology with Real-Time Detection. This research focuses on utilizing RT-DETR-X's real-time, end-to-end object identification capabilities to optimize drone technology for various applications such as military operations, geological investigation, and water conservation. RT-DETR-X achieves an impressive 54.8% Average Precision (AP) and 74 frames per second (FPS), outperforming comparable models in both speed and accuracy. The study investigates the VisDrone-DET dataset in detail, which includes a wide variety of tiny targets in scenes captured by UAV aerial photography. The dataset spans ten different categories, offering a strong foundation for thorough model testing. The research highlights the use of the original picture dataset for thorough training and assessment in addition to the useful use of the SAHI approach for improved small-scale object recognition. This research emphasizes the benefits of merging RT-DETR with the SAHI method through a thorough examination of the model's performance in several situations and a thorough investigation of the environmental setup. The results show that drone detection technologies have advanced significantly, providing a comprehensive foundation for successful and efficient aerial monitoring. In addition to increasing the model's detection accuracy, the integration creates novel opportunities for sophisticated picture analysis in UAV applications.

A technique for precisely estimating the position of road users in aerial pictures has been presented by Lu et al. [24]. Initially, oriented bounding boxes were used in conjunction with a deep learning-based technique to identify road users in aerial photos. After that, an error compensation plan was created to counteract the road user in order to achieve greater localization accuracy. This plan was based on an examination and modeling of the localization error caused by depth relief distortion. The effectiveness of the suggested strategy was assessed using field tests. The approach may help increase the legitimacy of UAVs in traffic applications, as the findings showed promising accuracy in locating road users.

3. PROPOSED WORK

This section delivers a brief explanation of the suggested model. The research utilizes three datasets, and Figure 1's input photographs are used to apply vehicle object identification.

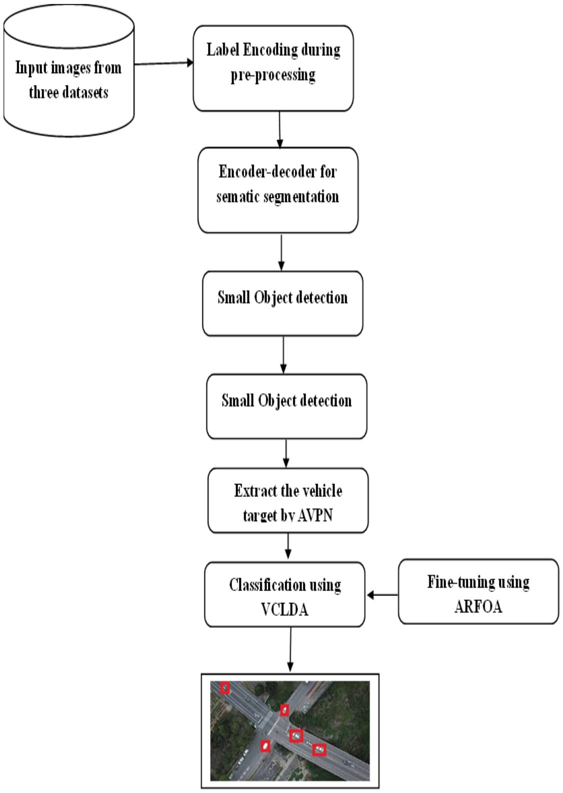
Input images from three datasets
Label Encoding during pre-processing
Encoder-decoder for sematic segmentation
Small Object detection
Small Object detection
Extract the vehicle target by AVPN
Classification using VCLDA
Fine-tuning using ARFOA

Figure 1: Workflow Of The Proposed Model

3.1. Datasets Description

The VAID, VEDAI, and DLR3K datasets are three sophisticated aerial imaging datasets that were taken into consideration by the study during the trials. Below are the specifics of these datasets:

3.1.1 VAID Dataset

For intelligent traffic monitoring by vehicle recognition and classification, H.Y. Lin et al. introduced the VAID [25] dataset in 2020. There were six thousand vehicle photographs in the collection, organized into seven minibuses, cement trailer. This footage was shot in a variety of lighting scenarios by use of a drone. For uniform vehicle photography, the drone is flown at an altitude of 90–95 meters. Images taken at a frame rate of 23.98 have a resolution of 2720 × 1530. The resolution of the pre-processed photos is 1137 × 640, and the photographs have been scaled. Ten locations in southern Taiwan's data collection conditions. Images show a variety of metropolitan settings, including a suburban area, a university campus, and a cityscape. The dataset's example photos are displayed in Figure 2.

Figure 2: Sample Images Of VAID Dataset

3.1.2 VEDAI Dataset

The VEDAI dataset was first suggested in 2015 [26]. Researchers may use the information to locate automobiles in aerial photos. Various properties, such as changing orientations, illumination, shadow, or obstructed objects, are displayed by the miniature cars in the collection. Additionally, a consistent technique is given so that other researchers may replicate and compare their results. We also provide the results of a few baseline methods for this dataset. Figure 3 shows a selection of photos taken from the VEDAI dataset.

Figure 3: Sample Images Of VEDAI Dataset

3.1.3 DLR-3K Dataset

The DLR-3K dataset [27] includes a variety of aerial views of automobiles in both urban and suburban settings. "Car" and "truck" are two of the vehicle categories included in the 20 high-resolution photos that make up the collection, which is also called the DLR Munich vehicle detection dataset. Images with the "car" class outnumber those with any other vehicle type. In order to prepare the model for use, the original photos are split into nine equal halves, yielding a grand total of 180 images. You can see some sample photos from the DLR3K collection in Figure 4.

Figure 4: Sample Image

3.2. Data preprocessing

The processing impact of the dataset has a direct correlation to the accuracy of semantic segmentation, and the network requires a substantial amount of time and energy to process the dataset prior to training. The preprocessing procedure and the consistent images in the dataset are organized as labelled graphs, where each target category is represented by a distinct color.

3.2.1. A label processing and encoding

One goal of label encoding is to establish a direct relationship between labels and colors. Using a 256-decimal-like function, it is necessary to store corresponding RGB values of all categories in a csv file to create a color map. Then, as demonstrated in formulas 1 and 2, hash mapping each pixel point in the color map to its corresponding category is performed.

$$k = (cm[0] \times 256 + cm[1]) \times 256 + cm[2] \quad (1)$$

$$cm2lbl[k] = i \quad (2)$$

Pixel RGB values are represented by cm[0], cm[20], besides cm[10]; the converted integer is denoted by k; cm2lbl is a hash table created using the hash purpose; and k is utilized as the pixel index in the cm2lbl category i that corresponds to the pixel.

3.3. Encoder-Decoder Network for Semantic Segmentation

Deep network topologies for semantic segmentation are extensively discussed in the computer vision field. The encoder-decoder design of the suggested paradigm is symmetrical [28]. The encoder relies on the VGG-16 model's convolutional layers. Using the ImageNet dataset for training, VGG-16 was created for the ILSRVC competition. Convolutional blocks, comprise both the encoder and the decoder. An encoder or a decoder, depending on whether it is a max pooling or unpooling layer, is thereafter supplied to each block. Dimension reduction and induction of translation invariance are accomplished by the use of the maximum pooling technique. The max pooling layer's twin operation, unpooling, takes the place of pooling in the decoder. The activations' value is moved into the mask of the extreme values ("argmax") calculated during the pooling step, and then through a skip link, straight into the decoder. Consecutive decoding convolutions densify the sparse activation map that is the outcome of such an upsampling. This enables the network to restore the input size to its original value by upsampling the feature activations from the decoder. As a result, the output feature maps retain the dimensions of the input. Consequently, the suggested network determines the value of each pixel independently. On smaller objects, the suggested model outperforms deconvolutional alternatives like DeconvNet in terms of accurately relocating abstract features to low-level saliency locations through the use of unpooling layers.

Our segmentation training set was constructed in the study by sliding a 128×128 px window over every 75% overlap, or a 32 px stride. The data is enhanced by this overlap. We employ every class from the ground truth in this experiment. In other words, we label each pixel with a class (such as "building" or "vehicle") and train the model to anticipate the vehicle mask. During testing, we apply a 50% overlap (i.e., a 64 px stride) to the tiles using a 128 × 128 px sliding window. To prevent a "mosaic" effect, we average overlapping forecasts to smooth them out at the window boundaries. Using the weights of a pre-trained VGG-16 on ImageNet, the study initializes the suggested encoder during training. Researchers concluded that half of the decoder's learning rate should be used for the encoder. We use Stochastic) for the network's training.

3.4. Small Object Detection

To prevent nearby automobiles from being merged into one blob, the proposed network's semantic maps should be precise enough, assuming the study would be conducted using VHR aerial photos that a human observer can differentiate cars on. Finding instances of vehicles in the pixel-level mask becomes as simple as extracting related components if this hypothesis is proven. After that, you may use the mask to regress the vehicle's bounding box. Nevertheless, the suggested network's predictions may contain noise because to CNN's fuzzy class transitions. So, to reduce disruptions in the network's predictions, we initially erode the vehicle mask by operating a morphological opening with a tiny radius. Secondly, in order to avoid erroneous vehicle classifications or false positives caused by segmentation artifacts, we remove items smaller than a certain threshold. This includes roof vents and street litter. This morphological opening, in conjunction with the linked extraction, is sufficient to accomplish efficient vehicle identification, despite its simplicity.

3.5. Accurate Vehicle Proposal Network (AVPN)

Using an AVPN that accepts a picture as input and produces a collection of vehicle-like areas 1 in the score, allowed us to reliably create all the vehicle- time. A fully convolutional network and its enhanced algorithm served as inspiration for our AVPN, which is based on an RPN [29]. To improve the AVPN's feature map for vehicle recognition, we used a combination of layers with varying resolutions. What follows is an explanation of AVPN's design and how it is trained.

3.5.1 Overall Architecture: Three fully linked layers and five convolutional layers make up the AVPN architecture. To create a concatenated feature map, we mutual the output feature maps of the final layers. To compute proposals, we added two more convolutional layers in place of the fully connected layers. The training images (of any size) are fed into the first convolutional layer (conv_1), which uses 96 kernels with a size of 7 × 7 × 3 to filter the input. The output of the previous convolutional layer is fed into the layer (conv_2), which filters it using 256 kernels measuring 5 × 5 × 96. Only after the first layers are configured are the rectified linear units and max _3, conv_4, and conv_5, which have 384 kernels of size 3 × 3 × 256, are connected to each other without the use of pooling or normalizing layers. In order to generate extra feature maps with 256, we layered a 3 × 3 convolutional layer on top of the conv_3 (conv_4) layer, namely conv_inter3 (conv_inter4), to merge multilevel feature maps with varied values. After applying local response normalization to the output of conv_inter3, conv_inter4, and conv_5, we fused the data into a single feature map cube, or hyper feature map. Our findings demonstrate that the concatenated map is complimentary for small-size vehicle identification since deeper levels are better suited for classification and shallower layers are more suited for localization.

We overlay the hyper feature map with a 3 × 3 window to create regions that resemble vehicles. It is then easy to construct the sliding process using conv_slid, a 3 × 3 convolutional layers. We can extract a 256-d feature vector for every sliding-window location, for a total of 256 feature maps. Next, this feature is input besides a two sibling 1 × 1 layer.

We simultaneously anticipate several areas associated with various aspect ratios and scales at each sliding-window location. We utilize three aspect ratios—3:2, 1:1, and 2:3—as well as three scales with box areas of 302, 402, and 502 pixels because the typical size of the vehicle is roughly 35 × 35 pixels. Every place can then forecast nine different kinds of areas. The predicted regions removed for AVPN training, and the remaining regions are given a binary class label (background or vehicle). We give a projected region a positive label if it has the maximum box. On the other hand, we label a forecast region negatively if its IoU ratio is less than a_n for every ground-truth box. The remaining sections are then thrown away. Here is how the IoU ratio is distinct.

$$a = \frac{aera(B_{vp} \cap B_{gt})}{aera(B_{vp} \cup B_{gt})} \quad (3)$$

where $aera\left(B_{vp} \cap B_{gt}\right)$ represents the connection of the vehicle ground truth box, and $aera\left(B_{vp} \cup B_{gt}\right)$ signifies their union.

3.5.2 Loss Function: We use a distinct loss purpose for the two-sibling output AVPN with the aforementioned definitions. For every projected region, the first sibling layer produces a vehicle-like score pc, which may be computed using a softmax classifier. The coordinates vecto are output by the second sibling layer $loc = (x, y, w, h)$ after the bounding-box regression, of each predicted region. The anticipated region's top-left coordinates are shown by x and y, while its width and height are indicated by w and h. In accordance with [29], we used a smooth L_1 loss layer to refine the organizes. Then for each positive labelled f^c and target ground-truth bounding-box loc^*, we accepted a multitask loss L_{AVPN} to box deterioration jointly:

$$L_{AVPN}(loc, f^c) = L_{cls}(f^c) + ap^* L_{bbr}(loc, loc^*) \quad (4)$$

where L_{cls} designates the classification of vehicle and background. p^* is label. If the region box is positive, $p^* = 1$, otherwise, $p^* = 0$, This indicates that boundingbox regression training is not affected by the backdrop. The parameter for balancing is α. We process a batch of training data throughout training, with each iteration having roughly the same amount of region boxes. In order to weight both, we set α = 2. L_{cls} and L_{bbr} terms equally. Moreover, L_{bbr} signifies a smooth L_1 loss defined as

$$L_{bbr}(loc, loc^*) = f_{L_1}(loc, loc^*),$$
$$where\ f_{L_1}(x) = \begin{cases} 0.5\,x^2 & if\ |x| < 1 \\ |x| - 0.5, & otherwise \end{cases} \quad (5)$$

3.5.3 Training AVPN: One way to train the AVPN is via stochastic gradient descent. To stop the first AVPN from being overfit, we used the classification for initialisation. We use a zero-mean Gaussian distribution with a standard deviation of 0.01 to randomly initialise the extra new convolutional layer weights. Each cycle ends with a parameter adjustment when we feed the network a new batch of labelled training data. Once the AVPN has completed its training, we use it together with an input aerial picture with pixels to generate around 300 candidate area boxes that are heavily overlapped. The recommended areas are subjected to non-maximum suppression (NMS) to reduce duplication, as determined by the vehicle confidence score. The next step is for the VCLDA to figure out which vehicle-like zones are important and in what directions.

3.6. Vehicle Classification Via Linear Discriminant Analysis (VCLDA)

One variation on the Bayesian concept is linear discriminant analysis [30]. Because it is a supervised technique, it requires class labels for training. LDA aims to maintain high inter-class variation and minimal intra-class variation. It is used to categorize the identified cars into different groups. Since LDA determines its coefficients based on the differences among the classes, scaling is not necessary. After separating each class, the following equation is used to combine the nine classes together:

$$\Sigma_b = \frac{1}{c}\Sigma_{i=1}^{c}(Meu_i - Meu)(Meu_i - Meu)^T \quad (6)$$

where classes C is represented by Meu_i, Σ characterizes the covariance, and Meu is means.

3.6.1. Fine-tuning of VCLDA using Artificial Root Foraging Optimization

To improve classification accuracy by optimizing LDA parameters, the research services the ARFOA model, which is described in the section below.

1) Classical Plant Root Growth Typical

The artificial root algorithm was designed based on the growth optimization technique. The lateral roots of a biological plant extend forth from the main root, while the major root of the plant moves toward the ground. In a similar vein, several lateral roots growing in different directions are also allowed for the lateral roots. The lateral roots can form in any direction with variable degrees of movement, but the primary roots are not allowed to do so. Thus, the traditional optimization model that forecasts plant root growth is also employed in the construction of the artificial method. It is believed that the characteristics of the soil impede root growth, and that the best remedies. For the VCLDA problems, the direction changes and length adjustments are thought to be the fine-tuning limits [31]. The following variables are taken into account for optimal plant growth, and the artificial model has done the same.

Factor 1: The concentration of auxin in the plants has a significant impact on the spatial arrangement of the roots. By looking at the issue, it enables the root to be routinely structured.

Factor 2: Children's root apices can be produced by a apex that grows in the same direction.

Factor 3: The root scheme produces a change of branches in response to auxin availability.

Factor 4: The main root's tip and the lateral roots' respective directions of movement along the trajectory are made possible via hydrotropism.

2) Auxin Regulation

When creating new branch count besides movement processes, the auxin concentration is the main parameter to consider. Soil nutrient availability is thus defined in the following way:

$$f_x = \frac{fitness_x - f_{low}}{f_{high} - f_{low}} \quad (7)$$

Precisely, the auxin attentiveness is written as

$$A_x = \frac{f_x}{\sum_{y=1}^{s} f_x} \quad (8)$$

where is $fitness_x$, f_x is the normalization fitness, f_{high} and flow characterize the existing root populace count besides s is the populace size.

3) Strategy on Main Root Growth

There is no branching or re-growing component to the main root's increasing likelihood. Based on the optimal individual operation derived from its present location, the main root's movement is determined. In mathematical notation, it is expressed as

$$I_x^t = I_x^{t-1} + l.\,\varepsilon.(I_{lbest} - I_x^{t-1}) \quad (9)$$

here, I_x^t implies a novel site, I_x^{t-1} depicts the spot where root x is located. In this context, l represents the learning inertia, ε is the unchanging chance coefficient ranging from 0 to 1, and I_{lbest} is the best separate currently located.

4) Branching Operator

The root apex estimations are used by the operator to produce a new individual. An estimate of concentration over the branch's included threshold value is used to predict it. A branch's potential offspring count is determined by

$$\begin{cases} branch\ individuals\ w_x & if\ A_x > threshold\ value \\ stop\ branching & otherwise \end{cases}$$
(10)

Therefore, the statistics of afresh produced apices are estimated from the subsequent equation

$$W_x = \varepsilon.A_x(B_{max} - B_{min}) + B_{min} \quad (11)$$

ε among 0 besides 1, A_x is the auxin attentiveness level at the root. B_{max} besides B_{min} describe the branched count. The site for emerging a novel branch root is foretold from distribution $N(I_x^t, \sigma^2)$. The written as

$$\sigma = \left(\frac{x_{max} - x}{x_{max}}\right)^2 \times (\sigma_{ini} - \sigma_{fin}) + \sigma_{fin} \quad (12)$$

where x_{max} is the extreme repetition, i current repetition index, σ_{ini} is the original standard deviation, besides σ_{fin} is the last normal deviation.

5) Lateral or Branch Root Development

In each feeding condition, the side roots are free to explore at accidental. As a result of these interdependent changes, the mathematical projection of the lateral roots' length and degree of growth is

$$I_x^t = I_x^{t-1} + \varepsilon(l_{max} D_i * \phi) \quad (13)$$

$$\phi = \frac{\delta_i}{\sqrt{\delta_i^T \times \delta_i}} \quad (14)$$

where l_{max} attitudes for the supreme length of the side root, D_i is the course of root i, besides ϕ attitudes for angle expressed with a accidental vector δ_i.

4. RESULTS AND ANALYSIS

System requirements for training the model include a GeForce RTX 3080 Ti GPU. The computational performance and correctness of the model are validated by parameter sensitivity analysis, which is used to establish the input layer size and other parameters. To calculate the training loss, we add all the squared errors from the last network layer. In Table 1 you can find the information of the parameters that were utilized for training.

Table 1. Limits used during the training of the perfect.

Vale/Range	Parameter Name
04	Mini-batch size
0.001	Degree of Learning (preliminary)
608X608	Input layer size
0.0005	Weight update relation
0.9	Worth of Momentum

We used an 80:20 split between the VEDAI and VAID training and test sets for developing the model. In contrast, the DLR-3K data set was split 70:30 between the train and test sets. Without using pretrained weights, the proposed model is trained over all datasets. During training, the suggested model ran 20,000 iterations on the VEDAI, VAID, and DLR-3K datasets. After every 5K rate is adjusted by a factor of 100. Every object has its own set of created bounding boxes. Based on the stated threshold, the suggested model selects the object with the highest IoU score.

4.1. Accuracy and Loss of planned model

Figure 5 and 6 shows the accuracy and loss of projected classifier model

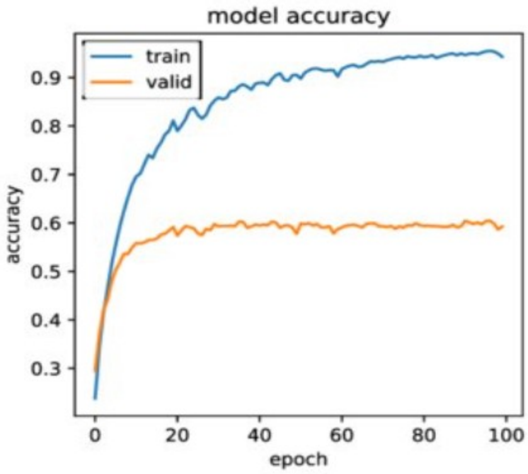

Figure 5: Accuracy On Training Besides Testing Data

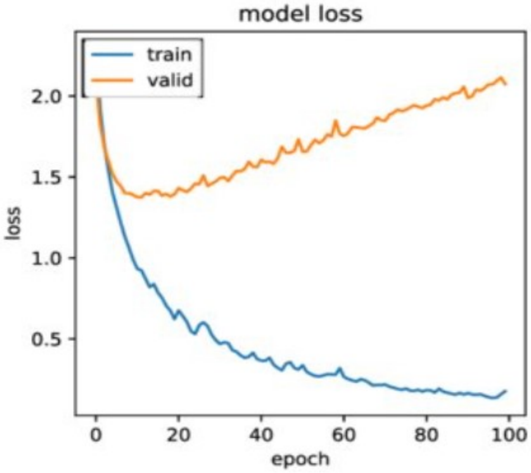

Figure 6: Loss On Training Besides Testing Data

4.2. Validation Investigation of Proposed Classifier

Table 2 presents the validation analysis of different learning rate on proposed VCLDA-ARFOA model.

Table 2: Experiment Analysis On Different Learning Rate On Three Datasets

Learning rate	Model	Sensitivity	Specificity	Accuracy	F1-Score	AUC–ROC
VAID						
0.1	Proposed	0.83	0.61	0.72	0.73	0.86
0.01		0.83	0.77	0.79	0.60	0.88
0.001		0.84	0.84	0.84	0.83	0.88
VEDAI						
0.1	Proposed	0.84	0.39	0.62	0.68	0.78
0.01		0.80	0.53	0.63	0.56	0.79
0.001		0.74	0.71	0.73	0.71	0.78
DLR-3K						
0.1	Proposed	0.73	0.49	0.61	0.61	0.70
0.01		0.69	0.58	0.63	0.54	0.73
0.001		0.69	0.58	0.64	0.62	0.74

In Table 2 above, the experiment analysis characterizes the performance of the proposed model with different learning rates on three datasets. For the VAID dataset, with a learning rate of 0.1, the proposed model achieved a sensitivity of 0.83, specificity of 0.61, accuracy of 0.72, F1-score of 0.73, and AUC-ROC of 0.86. With a learning rate of 0.01, the model attained a sensitivity of 0.83, specificity of 0.77, accuracy of 0.79, specificity of 0.60, and AUC-ROC of 0.88. At a learning rate of 0.001, the proposed model achieved a sensitivity of 0.84, specificity of 0.84, accuracy of 0.84, specificity of 0.83, and AUC-ROC of 0.88.

Moving to the VEDAI dataset, with a learning rate of 0.1, the proposed model attained a sensitivity of 0.84, specificity of 0.39, accuracy of 0.62, specificity of 0.68, and specificity of 0.78. At a learning rate of 0.01, the model achieved a sensitivity of 0.80, specificity of 0.53, accuracy of 0.63, accuracy of 0.56, and AUC-ROC of 0.79. With a learning rate of 0.001, the proposed model attained a sensitivity of 0.74, specificity of 0.71, accuracy of 0.71, specificity of 0.73, and AUC-ROC of 0.78.

For the DLR-3K dataset, at a learning rate of 0.1, the proposed model achieved a sensitivity of 0.73, accuracy of 0.49, accuracy of 0.61, specificity of 0.61, and AUC-ROC of 0.70. With a learning rate of 0.01, the model attained a sensitivity of 0.69, specificity of 0.58, accuracy of 0.63, specificity of 0.54, and AUC-ROC of 0.73. At a learning rate of 0.001, the projected model achieved a sensitivity of 0.69, specificity of 0.58, accuracy of 0.64, specificity of 0.62, and AUC-ROC of 0.74.

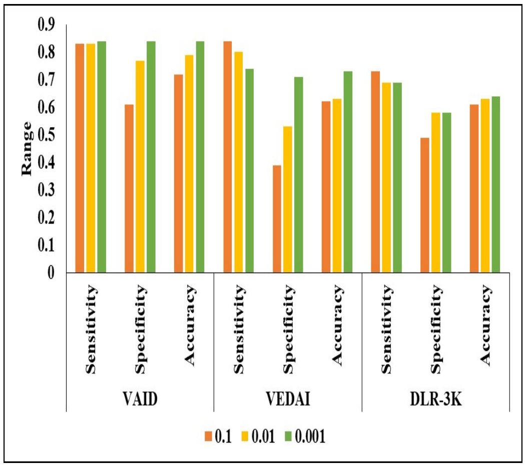

Figure 7: Graphical Description Of Proposed Model On Different Learning Rate

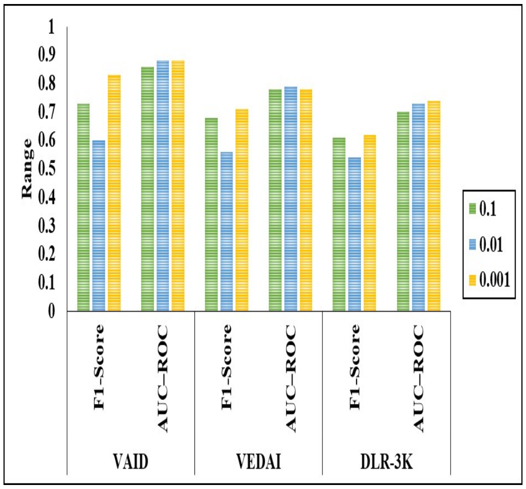

Figure 8: Visual Representation Of Proposed Model On Three Datasets.

4.3. Comparative Analysis of Proposed classical

Table 3 mentions the comparative study of proposed model with existing procedures, where the techniques are implemented with three datasets and average results are mentioned on whole three datasets.

Table 3: Comparative Inspection Of Projected Classical With Existing Procedures

Model	Accuracy	Precision	Recall	F1
FRCNN	0.9422	0.9111	0.7332	0.8033
ACF	0.9314	0.9515	0.6138	0.7043
HRPN	0.9425	0.9112	0.7335	0.8034
CPPM [32]	0.9698	0.9698	0.9698	0.9698
YOLO [17]	0.9696	0.9696	0.9696	0.9696
LP-NMS [19]	0.9684	0.9684	0.9684	0.9684
DLSTM [20]	0.9675	0.9675	0.9675	0.9675
Aero-YOLO [22]	0.9720	0.9720	0.9720	0.9720
VCLDA-ARFOA	0.9823	0.9823	0.9823	0.9823

In Table 3 above, the Proportional comparison of Predictable perfection with existing procedures is presented. In the analysis, the FRCNN technique achieved an accuracy of 0.9422, precision of 0.9111, recall of 0.7332, and F1-score of 0.8033 consistently. The ACF technique attained an accuracy of 0.9314, precision of 0.9515, recall of 0.6138, and F1-score of 0.7043 correspondingly. The HRPN technique achieved an accuracy of 0.9425, precision of 0.9112, recall of 0.7335, and F1-score of 0.8034 correspondingly. The CPPM [32] technique attained an accuracy, precision, recall, and F1-score of 0.9698 consistently. The YOLO [17] technique attained an accuracy, precision, recall, and F1-score of 0.9696 similarly. The LP-NMS [19] technique achieved an accuracy, precision, recall, and F1-score of 0.9684 correspondingly. The DLSTM [20] technique achieved an accuracy, precision, recall, and F1-score of 0.9675 correspondingly. The Aero-YOLO [22] technique attained an accuracy, precision, recall, and F1-score of 0.9720 correspondingly. Finally, the VCLDA-ARFOA technique achieved an accuracy, precision, recall, and F1-score of 0.9823 consistently.

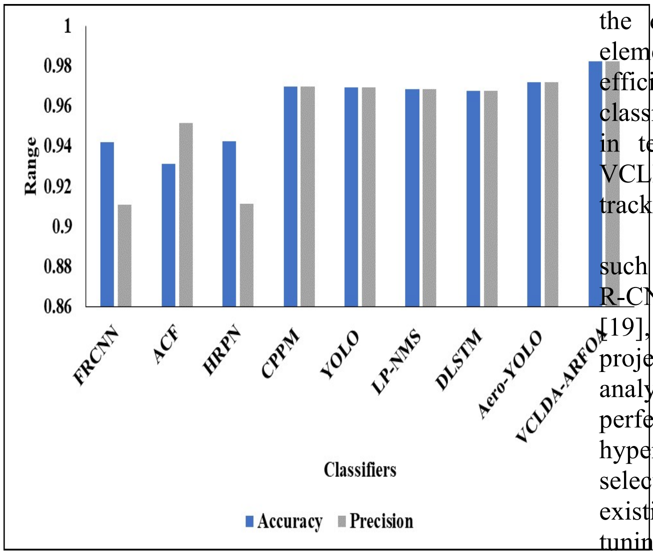

Figure 9: Comparison Analysis Of Different Classifiers

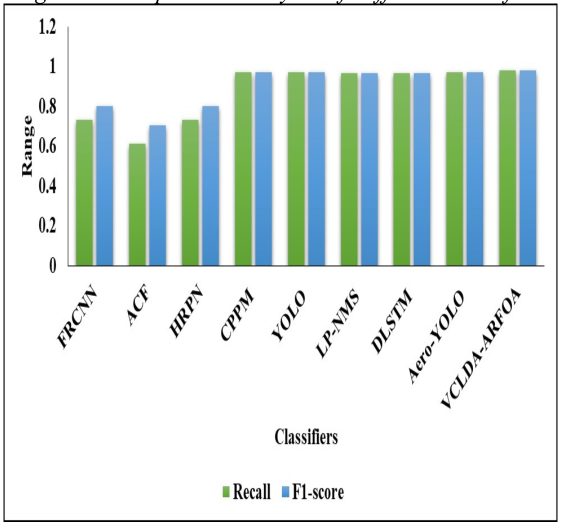

Figure 10: Visual Analysis Of Various DL Classifiers For Vehicle Detection

4.4. Discussion

Using high-resolution aerial imagery, the planned traffic monitoring system aims to regulate traffic. In this research, we built a system that can accurately isolate automobiles in aerial photos by using convolutional neural network (CNN)-based semantic segmentation. The suggested VCLDA-ARFOA further examines these split objects for vehicle detection. The cars that have been spotted are subsequently classified into various groups. The primary requirement, especially for vehicle recognition, is high-resolution aerial photos. Consequently, to achieve notable outcomes for segmented scene photos, an efficient CNN-based segmentation method was integrated. Analyzing the segmented aerial photos allows for the identification of various vehicles. An innovative VCLDA-ARFOA technique is developed during the detection stage, which is the most crucial element of the scheme, to improve overall efficiency. When it comes to detection and classification, VCLDA-ARFOA excels particularly in terms of recall. Additionally, the suggested VCLDA-ARFOA method enhances the successful tracking of detected vehicles.

The research work applied various systems such as region suggestion network (HRPN), Faster R-CNN, aggregated CPPM, YOLO [17], LP-NMS [19], DLSTM [20], and Aero-YOLO [22] to our projected VEDAI and DLR-3k datasets. From this analysis, it is visibly shown that the projected perfect achieved better performance because the hyperparameter tuning of LDA is optimally selected by the ARFOA model. However, the existing models do not focus on hyperparameter tuning, leading to average performance, as graphically shown in Figures 9 and 10.

4.5 Open Issues and Limitations

Addressing the following open issues and limitations would not only strengthen the research findings but also contribute to the development of more robust and practical solutions for intelligent vehicle detection and classification in aerial imagery.

Scalability and Real-Time Processing: While the proposed system shows promise in vehicle detection and classification, its scalability to real-time processing and its efficiency in handling large-scale aerial imagery remain unclear. Real-time processing is crucial for applications like traffic monitoring and management, where timely responses are necessary.

Robustness to Environmental Variability: The research mentions challenges arising from dynamic scenes with inaccurate vehicle information and cluttered backgrounds. However, further investigation into the system's robustness to various environmental conditions, such as weather changes (e.g., fog, rain, snow) and lighting variations (e.g., day vs. night), would be valuable. Ensuring the system's performance across different environmental conditions is essential for its practical deployment.

Generalization to Different Geographical Regions: The datasets used in the experiments cover a broad spectrum of backgrounds and vehicle types in urban and rural settings worldwide. However, the generalization of the proposed system to different geographical regions with distinct characteristics (e.g., infrastructure, vehicle types, traffic regulations) remains to be thoroughly explored. Adapting the system to specific regional nuances could improve its performance and applicability in diverse contexts.

Evaluation on Additional Benchmarks: While the experimental results demonstrate the efficacy of the proposed approach on the VAID, VEDAI, and DRL3K datasets, evaluation on additional benchmark datasets would provide a more comprehensive assessment of its performance. Utilizing datasets with different characteristics and challenges could offer insights into the system's strengths and weaknesses in varying scenarios.

Ethical and Privacy Considerations: As with any surveillance system, ethical and privacy considerations are paramount. Further discussion on how the proposed system addresses or mitigates potential concerns related to privacy invasion, data security, and unintended consequences (e.g., biases in decision-making) would be essential for its responsible deployment.

5. CONCLUSION

In our research, we present a robust system designed to identify vehicles in drone aerial photos, addressing crucial areas such as smart surveillance systems, intelligent traffic monitoring, and efficient traffic management. Leveraging the proposed LDA model, our innovative traffic monitoring scheme significantly enhances the effectiveness of vehicle detection. Initially, our methodology employs an encoder-decoder module to efficiently segment aerial images before precisely identifying various vehicles. These vehicles are then categorized using linear discriminant analysis, followed by optimization of hyperparameters through the ARFOA model. Experimental validation on the VAID, VEDAI, and DRL3K datasets demonstrates the efficacy of our approach, surpassing previous state-of-the-art methods.

The datasets used in our trials are dynamic, diverse, and complex, encompassing a broad spectrum of backgrounds and vehicle types in urban and rural settings worldwide. Our proposed detection module, VCLDA-AROA, exhibits varying degrees of success across datasets due to the dynamic nature of the scenes, which may contain inaccurate vehicle information and cluttered backgrounds. Challenges arose in settings where objects were partially or completely obscured, such as when obscured by trees or overshadowed by nearby buildings.

Moving forward, we aim to further enhance traffic surveillance using deep learning techniques, focusing on improving vehicle recognition and tracking. By addressing these challenges, we strive to make vehicle tracking more accurate and effective in diverse and challenging environments.

REFERENCES

[1] Kumar, S., Jain, A., Rani, S., Alshazly, H., Idris, S. A., & Bourouis, S. (2022). Deep Neural Network Based Vehicle Detection and Classification of Aerial Images. Intelligent Automation & Soft Computing, 34(1).

[2] Sabour, M. H., Jafary, P., & Nematiyan, S. (2023). Applications and classifications of unmanned aerial vehicles: A literature review with focus on multi-rotors. The Aeronautical Journal, 127(1309), 466-490.

[3] Gupta, P., Pareek, B., Singal, G., & Rao, D. V. (2022). Edge device based military vehicle detection and classification from UAV. Multimedia Tools and Applications, 81(14), 19813-19834.

[4] Park, G., Park, K., Song, B., & Lee, H. (2022). Analyzing impact of types of UAV-derived images on the object-based classification of land cover in an urban area. Drones, 6(3), 71.

[5] Phalguna Krishna E S, Venkata Nagaraju Thatha, Gowtham Mamidisetti, Srihari Varma Mantena, Phanikanth Chintamaneni, Ramesh Vatambeti. Hybrid deep learning model with enhanced sunflower optimization for flood and earthquake detection, Heliyon, Volume 9, Issue 10, 2023, e21172.

[6] Behera, T. K., Bakshi, S., Sa, P. K., Nappi, M., Castiglione, A., Vijayakumar, P., & Gupta, B. B. (2023). The NITRDrone dataset to address the challenges for road extraction from aerial images. Journal of Signal Processing Systems, 95(2), 197-209.

[7] Bouguettaya, A., Zarzour, H., Kechida, A., & Taberkit, A. M. (2022). Deep learning techniques to classify agricultural crops through UAV imagery: A review. Neural

Computing and Applications, 34(12), 9511-9536.

[8] Guo, Q., Zhang, J., Guo, S., Ye, Z., Deng, H., Hou, X., & Zhang, H. (2022). Urban tree classification based on object-oriented approach and random forest algorithm using unmanned aerial vehicle (uav) multispectral imagery. Remote Sensing, 14(16), 3885.

[9] Chakravarthy, A. S., Sinha, S., Narang, P., Mandal, M., Chamola, V., & Yu, F. R. (2022). Dronesegnet: Robust aerial semantic segmentation for uav-based iot applications. IEEE Transactions on Vehicular Technology, 71(4), 4277-4286.

[10] Thirumalraj, A., Asha, V., & Kavin, B. P. (2023). An Improved Hunter-Prey Optimizer-Based DenseNet Model for Classification of Hyper-Spectral Images. In AI and IoT-Based Technologies for Precision Medicine (pp. 76-96). IGI global.

[11] Behera, T. K., Bakshi, S., Nappi, M., & Sa, P. K. (2023). Superpixel-based multiscale CNN approach toward multiclass object segmentation from UAV-captured aerial images. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 16, 1771-1784.

[12] Khidher, A. M., & Sehree, N. A. (2022). Automatic trees density classification using deep learning of unmanned aerial vehicles images. International Journal of Mechanical Engineering, 7(2), 3155-3164.

[13] Jia, J., Cui, W., & Liu, J. (2022). Urban catchment-scale blue-green-gray infrastructure classification with unmanned aerial vehicle images and machine learning algorithms. Frontiers in Environmental Science, 9, 778598.

[14] Lin, H. Y., Tu, K. C., & Li, C. Y. (2020). Vaid: An aerial image dataset for vehicle detection and classification. IEEE Access, 8, 212209-212219.

[15] Ozdemir, A., & OZKAN, I. A. (2023, September). Classification of Unmanned Aerial Vehicle and Bird Images Using Deep Transfer Learning Methods. In Proceedings of the International Conference on Advanced Technologies (Vol. 11, pp. 189-196).

[16] Kumawat, H. C., Chakraborty, M., & Raj, A. A. B. (2022). DIAT-RadSATNet—A novel lightweight DCNN architecture for micro-Doppler-based small unmanned aerial vehicle (SUAV) targets' detection and classification. IEEE Transactions on Instrumentation and Measurement, 71, 1-11.

[17] Orić, M., Galić, V., & Novoselnik, F. (2024). Synthetic car dataset for vehicle detection: Integrating aerial and satellite imagery. Data in brief, 110105.

[18] Mustafa, N. E., & Alizadeh, F. (2024). Unmanned Aerial Vehicle (UAV) Images of Road Vehicles Dataset. Data in Brief, 110264.

[19] Wang, H., Wang, C., Fu, Q., Zhang, D., Kou, R., Yu, Y., & Song, J. (2024). Cross-Modal Oriented Object Detection of UAV Aerial Images Based on Image Feature. IEEE Transactions on Geoscience and Remote Sensing, 62, 1-21.

[20] Vaiyapuri, T., Sivakumar, M., Shridevi, S., Parvathy, V. S., Ramesh, J. V. N., Syed, K., & Mohanty, S. N. (2024). An intelligent water drop algorithm with deep learning driven vehicle detection and classification. AIMS Mathematics, 9(5), 11352-11371.

[21] Sun, H., Guo, J., Meng, Z., Zhang, T., Fang, J., Lin, Y., & Yu, H. (2024). EVD4UAV: An Altitude-Sensitive Benchmark to Evade Vehicle Detection in UAV. arXiv preprint arXiv:2403.05422.

[22] Shao, Y., Yang, Z., Li, Z., & Li, J. (2024). Aero-YOLO: An Efficient Vehicle and Pedestrian Detection Algorithm Based on Unmanned Aerial Imagery. Electronics, 13(7), 1190.

[23] Muzammul, M., Algarni, A. M., Ghadi, Y. Y., & Assam, M. (2024). Enhancing UAV Aerial Image Analysis: Integrating Advanced SAHI Techniques with Real-Time Detection Models on the VisDrone Dataset. IEEE Access.

[24] Lu, L., & Dai, F. (2024). Accurate road user localization in aerial images captured by unmanned aerial vehicles. Automation in Construction, 158, 105257.

[25] H.-Y. Lin, K.-C. Tu, and C.-Y. Li, ''VAID: An aerial image dataset for vehicle detection and classification,'' IEEE Access, vol. 8, pp. 212209–212219, 2020.

[26] S. Razakarivony and F. Jurie, ''Vehicle detection in aerial imagery: A small target detection benchmark,'' J. Vis. Commun. Image Represent., vol. 34, pp. 187–203, Jan. 2016.

[27] K. Liu and G. Mattyus, ''Fast multiclass vehicle detection on aerial images,'' IEEE Geosci. Remote Sens. Lett., vol. 12, no. 9, pp. 1938–1942, Sep. 2015.

[28] Baswaraju, S., Maheswari, V.U., Chennam, k.K. et al. Future Food Production Prediction Using AROA Based Hybrid Deep Learning Model in Agri-Sector. Hum-Cent Intell Syst 3,

521–536 (2023). https://doi.org/10.1007/s44230-023-00046-y

[29] Wang, H., & Xiao, N. (2023). Underwater object detection method based on improved Faster RCNN. Applied Sciences, 13(4), 2746.

[30] Li, S., Zhang, H., Ma, R., Zhou, J., Wen, J., & Zhang, B. (2023). Linear discriminant analysis with generalized kernel constraint for robust image classification. Pattern Recognition, 136, 109196.

[31] Liu, Y., Liu, J., Ma, L., & Tian, L. (2017). Artificial root foraging optimizer algorithm with hybrid strategies. Saudi Journal of Biological Sciences, 24(2), 268-275.

[32] Rafique, A. A., Al-Rasheed, A., Ksibi, A., Ayadi, M., Jalal, A., Alnowaiser, K., ... & Park, J. (2023). Smart traffic monitoring through pyramid pooling vehicle detection and filter-based tracking on aerial images. IEEE Access, 11, 2993-3007.