



ARTICLE

Remote Sensing Imagery for Multi-Stage Vehicle Detection and Classification via YOLOv9 and Deep Learner

Naif Al Mudawi^{1,*}, Muhammad Hanzla², Abdulwahab Alazeb¹, Mohammed Alshehri¹,
Haifa F. Alhasson³, Dina Abdulaziz AlHammadi⁴ and Ahmad Jalal^{2,5}

¹Department of Computer Science, College of Computer Science and Information System, Najran University, Najran, 55461, Saudi Arabia

²Department of Computer Science, Air University, Islamabad, 44000, Pakistan

³Department of Information Technology, College of Computer, Qassim University, Buraydah, 52571, Saudi Arabia

⁴Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia

⁵Department of Computer Science and Engineering, College of Informatics, Korea University, Seoul, 02841, Republic of Korea

*Corresponding Author: Naif Al Mudawi. Email: naalmudawi@nu.edu.sa

Received: 14 March 2025; Accepted: 27 May 2025; Published: 30 July 2025

ABSTRACT: Unmanned Aerial Vehicles (UAVs) are increasingly employed in traffic surveillance, urban planning, and infrastructure monitoring due to their cost-effectiveness, flexibility, and high-resolution imaging. However, vehicle detection and classification in aerial imagery remain challenging due to scale variations from fluctuating UAV altitudes, frequent occlusions in dense traffic, and environmental noise, such as shadows and lighting inconsistencies. Traditional methods, including sliding-window searches and shallow learning techniques, struggle with computational inefficiency and robustness under dynamic conditions. To address these limitations, this study proposes a six-stage hierarchical framework integrating radiometric calibration, deep learning, and classical feature engineering. The workflow begins with radiometric calibration to normalize pixel intensities and mitigate sensor noise, followed by Conditional Random Field (CRF) segmentation to isolate vehicles. YOLOv9, equipped with a bi-directional feature pyramid network (BiFPN), ensures precise multi-scale object detection. Hybrid feature extraction employs Maximally Stable Extremal Regions (MSER) for stable contour detection, Binary Robust Independent Elementary Features (BRIEF) for texture encoding, and Affine-SIFT (ASIFT) for viewpoint invariance. Quadratic Discriminant Analysis (QDA) enhances feature discrimination, while a Probabilistic Neural Network (PNN) performs Bayesian probability-based classification. Tested on the Roundabout Aerial Imagery (15,474 images, 985K instances) and AU-AIR (32,823 instances, 7 classes) datasets, the model achieves state-of-the-art accuracy of 95.54% and 94.14%, respectively. Its superior performance in detecting small-scale vehicles and resolving occlusions highlights its potential for intelligent traffic systems. Future work will extend testing to nighttime and adverse weather conditions while optimizing real-time UAV inference.

KEYWORDS: Feature extraction; traffic analysis; unmanned aerial vehicles (UAV); you only look once version 9 (YOLOv9); machine learning; remote sensing for traffic monitoring; computer vision

1 Introduction

Unmanned Aerial Vehicles (UAVs) have become vital tools across various fields, including surveillance, disaster management, traffic monitoring, and urban planning [1]. Their flexibility enables the capture of high-resolution visual data from diverse angles and altitudes for real-time analysis. UAV-based imaging offers clear advantages over ground-based methods, such as a broader field of view, better spatial resolution, and greater



adaptability [2]. Compared to satellite imagery, UAV data is more cost-effective, provides timely updates, and supports real-time collection, making it ideal for dynamic tasks like traffic surveillance and vehicle detection. This application has gained importance in Intelligent Transportation Systems (ITS), traffic analysis, and urban planning [3,4]. However, challenges like object scale variation, occlusion, and computational limits require more advanced detection techniques. Aerial image classification is also crucial in applications like traffic analysis, disaster response, and city planning. Its complexity stems from varying object scales, occlusions, and environmental conditions. Deep learning, especially convolutional neural networks (CNNs), effectively handles these issues by extracting hierarchical features, improving accuracy and robustness over traditional methods. Leveraging deep learning, our model ensures efficient and accurate classification across diverse aerial scenarios. Traditional vehicle detection techniques, such as sliding-window searches and shallow feature extraction, often fall short in meeting the demands of complex aerial imagery. Sliding-window methods require exhaustive scanning across multiple scales and positions, resulting in prohibitive computational costs for high-resolution UAV imagery. Similarly, shallow learning techniques (e.g., Haar cascades, Histogram object gradients and Support vector machine (HOG-SVM)) rely on handcrafted features that lack robustness to scale variations, occlusions, and lighting changes, leading to poor generalization in dynamic aerial environments [5]. The advent of CNNs has significantly advanced object detection in aerial images by automating hierarchical feature extraction, enabling a robust representation of objects across scales and orientations. Recent studies conducted extensive analyses of the YOLO algorithm family, particularly YOLOv8, highlighting its advancements in detection precision, inference speed, and real-time applicability for Intelligent Transportation Systems (ITS). For instance, Bakirci [6] demonstrated that YOLOv8's decoupled head structure and C2f module improve accuracy by 18% over YOLOv5 in aerial drone imagery while maintaining faster inference speeds. Similarly, Bakirci [7] validated the compact YOLOv8n model's suitability for ITS, leveraging its C3 modules to balance computational efficiency with detection robustness. However, these studies also identified critical limitations in YOLOv8, including persistent misclassifications due to vehicle shape variations, lighting inconsistencies, and occlusion challenges exacerbated in UAV-based aerial imagery where objects are often small, densely packed, or partially obscured.

This study presents a six-stage framework for vehicle detection and classification in aerial imagery, combining radiometric calibration, CRF segmentation, YOLOv9 detection, and feature extraction using MSER, BRIEF, and Affine-SIFT. QDA optimizes features, while PNN handles classification. Evaluated on three benchmark datasets, the model achieves high accuracy and outperforms traditional and CNN-based approaches:

The main contributions of our system are listed below:

- The model integrates radiometric calibration and Conditional Random Field (CRF) segmentation to reduce noise and enhance brightness before the detection phase, reducing model complexity.
- The use of YOLOv9 improves vehicle detection accuracy, especially for objects of varying sizes in aerial images.
- Multiple features such as MSER, BRIEF, and Affine-SIFT are extracted, providing robust, scale-invariant, and rotation-invariant descriptors for better vehicle classification.
- Optimized feature representation via Quadratic Discriminant Analysis (QDA) and classification using Probabilistic Neural Network (PNN) significantly enhances accuracy in aerial vehicle detection, effectively handling image variability.

The rest of this article is structured as follows: [Section 2](#) gives an examination of relevant work and current techniques. [Section 3](#) covers the architecture of the proposed system. [Section 4](#) offers experimental setup and performance assessment, followed by the discussion in [Section 5](#). Finally, [Section 6](#) concludes the study and suggests suggestions for further research.

2 Literature Review

Vehicle detection and classification in aerial imagery are vital for traffic management, urban planning, and surveillance. This section reviews key systems, highlighting their methodologies, advancements, and performance metrics.

2.1 Traditional Methods

Liu et al. [8] introduce a robust vehicle detection scheme that addresses issues such as overhead perspectives and complex backgrounds. The proposed algorithm generates oriented proposals, enclosing vehicle objects as rotated rectangles to better fit their orientation, but limitations include the two-stage detection process, which can be complex and may require significant computational resources, potentially limiting real-time application feasibility. Hamadi et al. [9] introduced an automated system for UAV detection and classification using ground-based cameras. Their method utilizes Histogram of Oriented Gradients (HOG) features to transform observed UAVs into a 2D feature space, facilitating precise class separation. This system demonstrates high accuracy and reliability in distinguishing UAVs from other objects, though its effectiveness is influenced by environmental factors such as lighting and background complexity.

2.2 Machine Learning-Based Methods

Arinaldi et al. [10] proposed two vehicle detection methods: one using a Mixture of Gaussian (MoG) for background modeling with SVM for classification. Though efficient, it struggled with occlusions and dense traffic. Kumar et al. [11] introduced a deep neural network-based approach, achieving 92.06% accuracy on the Vehicle Detection in Aerial Imagery (VEDAI) dataset but noted limitations in real-time performance, requiring further optimization. Goecks et al. [12] explored deep learning-based fusion of visible and Long-Wave Infrared (LWIR) imagery for sUAS detection, achieving a 71.2% detection rate with a 2.7% false alarm rate. While fusion improved detection, challenges remained in distinguishing sUAS from other heat sources. Lee [13] used synthetic UAV data with traditional ML models like Decision Trees and K-Nearest Neighbors (KNN), revealing that detection accuracy decreased in nadir views and with varied object poses, highlighting the need for adaptable models.

2.3 Deep Learning-Based Methods

Li et al. [14] proposed a unified framework for vehicle detection and counting, using a scale-adaptive anchor generator and feature pyramid to improve multi-scale detection. While efficient, the method faces real-time computational challenges. Isaac-Medina et al. [15] benchmarked deep neural networks for UAV detection and tracking, achieving a mAP of 82.8% on infrared data, highlighting the challenges of cross-modality evaluation. Deng [16] introduced a vehicle-proposal network (AVPN) with hierarchical feature layers, excelling in small vehicle detection but occasionally missing detections in dense traffic. Tan [17] developed a CNN-based vehicle categorization method using aerial images, leveraging motion changes and feature matching, but it struggles with cluttered backgrounds and high computational demand. Jiang [18] proposed a parallel neural network combining ICA-2D-CNN and 3D-CNN for vehicle classification, enhancing feature extraction but suffering from noise interference and boundary imprecision. Zhou et al. [19] improved vehicle detection in urban traffic by integrating EfficientViT into YOLOv8n's backbone, achieving a lightweight design for resource-constrained devices. They added a Convolutional Block Attention Module (CBAM) in the neck and replaced convolutional layers with GhostConv in the head, reducing parameters while maintaining speed. Experimental results showed a precision of 91.17%, an mAP@0.5 of 75.45%, and a recall of around 70.01%, outperforming YOLOv8n in complex traffic scenarios. Carion et al. [20] introduced DETR (DEtection TRansformer), replacing CNN pipelines with transformer architectures, achieving 92.8% mAP

on aerial datasets but facing impractical quadratic complexity for real-time UAV use. Tan et al. [21] enhanced scalability with compound scaling and Bidirectional Feature Pyramid Network (BiFPN), achieving 93.6% mAP on UAV benchmarks. However, EfficientDet lacks affine invariance, addressed in our work through ASIFT-based feature extraction. While these models excel in general object detection, they struggle with UAV-specific challenges like scale, viewpoint changes, and occlusions. Our hybrid framework overcomes this, combining YOLOv9 with classical descriptors (MSER, BRIEF, ASIFT) and QDA optimization, achieving 95.54% accuracy on the Roundabout dataset, outperforming YOLOv8 and EfficientDet by 1.3%–2.7%.

3 Materials and Methods

The proposed system for aerial vehicle detection and classification consists of six phases. It begins with radiometric calibration to correct brightness inconsistencies. CRF segmentation separates vehicles from the background, and YOLOv9 detects vehicles accurately and swiftly. Feature extraction using MSER, BRIEF, and ASIFT follows, ensuring robust recognition. QDA optimizes features, reducing dimensionality while retaining key information. Finally, a PNN classifies vehicles into predefined categories, ensuring high accuracy across diverse datasets. Fig. 1 shows the architecture workflow of the proposed system.

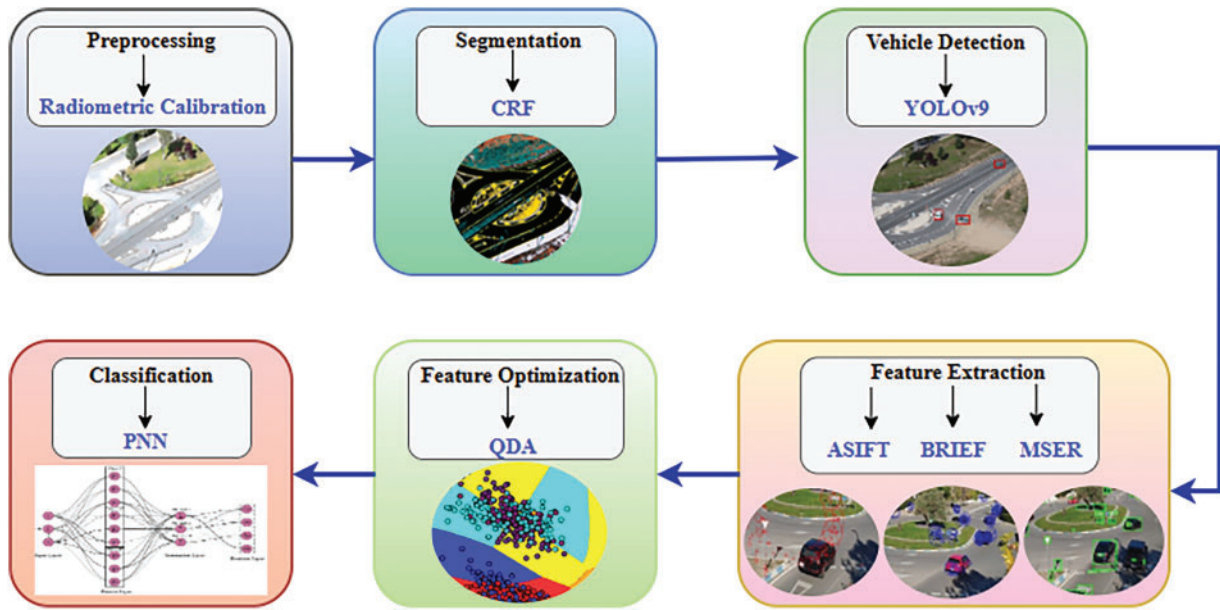


Figure 1: Architecture of the proposed intelligent traffic surveillance system

3.1 Image Preprocessing via Radiometric Calibration

Preprocessing is crucial for vehicle detection in aerial images. Our method uses radiometric calibration to correct distortions caused by sensor limitations, environmental factors, and lighting variations. This procedure ensures pixel values accurately reflect the surface's reflectance, essential for effective detection and classification [22]. The process adjusts pixel intensities to match real light detected by the sensor, compensating for noise and issues like vignetting, sensor upgrades, and ambient interference. Radiometric calibration is defined in Eq. (1):

$$I_{calibrated}(x, y) = \frac{I_{raw}(x, y) - I_{dark}(x, y)}{E(x, y) \cdot T(x, y)} \quad (1)$$

where $I_{calibrated}$ is the calibrated pixel intensity at (x, y) , $I_{raw}(x, y)$ the raw pixel intensity, $I_{dark}(x, y)$ the sensor noise, $E(x, y)$ the exposure time, and $T(x, y)$ the transmission loss from atmospheric effects. Radiometric calibration also adjusts for the sensor's response function, typically modeled as in Eq. (2):

$$L_{calibrated} = \frac{I_{raw}}{I_{Gain}} + Offset \quad (2)$$

where $L_{calibrated}$ is the calibrated luminance, and $Gain$ and $Offset$ represent the camera's gain factor and offset values, respectively. The final outcome of the preprocessing can be depicted in Fig. 2.



Figure 2: Enhanced preprocessing results using radiometric calibration on aerial images

3.2 Image Segmentation via Conditional Random Field (CRF)

Segmentation is crucial in vehicle detection, and we use Conditional Random Fields (CRF) to precisely separate vehicles from the background in aerial images. CRF, a probabilistic graphical model, assigns labels to pixels while considering contextual dependencies, ensuring spatial coherence and smooth boundaries [23]. It refines initial detection results by modeling the conditional probability of label assignments based on image features. The objective is to maximize correct labeling across the image, as formulated in Eq. (3):

$$P(Y|X) = \frac{1}{Z(X)} \exp\left(-\sum_{c \in C} \psi_c(Y_c, X_c)\right) \quad (3)$$

where $P(Y|X)$ is the conditional probability of label configuration Y (e.g., vehicle or background) given features X , with $Z(X)$ ensuring the probability sums to 1, and C representing cliques. The potential function $\psi_c(Y_c, X_c)$ includes unary potential (label likelihood from pixel features) and pairwise potential, as shown in Eq. (4).

$$\psi_u(Y_i, X_i) = -\log P(Y_i|X_i) \quad (4)$$

where $P(Y_i|X_i)$ is the likelihood of pixel i being assigned label Y_i based on its observed features X_i . The pairwise potential, on the other hand, encourages spatial smoothness by penalizing label changes between neighboring pixels, ensuring that adjacent pixels with similar features are assigned the same label. The results of segmentation can be seen in Fig. 3.

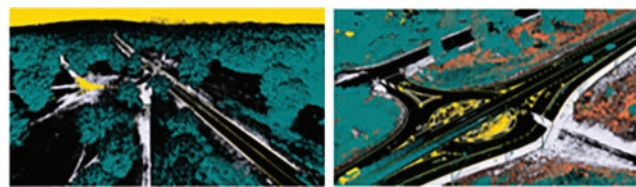


Figure 3: Segmentation using CRF over the aerial images

3.3 Vehicle Detection via YOLOv9

YOLOv9 powers the vehicle detection module, using its hierarchical convolutional transformer backbone for multi-scale feature extraction and BiFPN for optimal feature aggregation. Its anchor-free mechanism and attention-weighted modules refine bounding box predictions, improving classification in dense scenes. The detection process starts with image segmentation, followed by YOLOv9's dynamic keypoint-based bounding box regression, reducing computational overhead [24]. The attention module enhances spatial focus, making YOLOv9 highly efficient for real-time detection in diverse aerial imagery. The framework optimizes performance using a weighted loss function that balances classification, bounding box regression, and objectness scores as defined in Eq. (5).

$$L = \lambda_{cls} L_{cls} (C, \hat{C}) + \lambda_{bpx} L_{box} (B, \hat{B}) + \lambda_{obj} L_{obj} (S, \hat{S}) \quad (5)$$

where L_{cls} is the classification loss (binary cross-entropy), λ_{bpx} is the bounding box regression loss (smooth L1 or GIoU), and L_{obj} measures object confidence. YOLOv9 optimizes with AdamW or Layer-wise Adaptive Moments optimizer for Batch training (LAMB), fine-tuning hyperparameters. Its anchor-free approach enhances efficiency. YOLOv9 parameters are in Table 1, and detection output is in Fig. 4. YOLOv9 is employed as the detection backbone due to its recent architectural advancements, such as the Generalized Efficient Layer Aggregation Network (GELAN) and Programmable Gradient Information (PGI) module, which improve detection performance across varying object scales, especially important for aerial views. Compared to earlier YOLO versions, YOLOv9 offers improved precision and speed, making it well-suited for real-time aerial vehicle detection.

Table 1: Parameters used during the training of the model

Parameter name	Values
Learning rate	0.001
Epochs	80
IoU threshold	0.5
Batch size	32
Momentum	0.9
Confidence threshold	0.25
Weight decay	0.0005
Activation function	LeakyReLU
Input layer size	640×640
Weight decay	0.0001
Optimizer	AdamW



Figure 4: Vehicle detection using the YOLOv9 algorithm

To further substantiate our choice of YOLOv9 as the detection backbone, we conducted a comparative analysis against recent YOLO versions. Table 2 presents the detection performance (Precision, Recall, F1score) along with model complexity (number of parameters and FLOPs) and processing speed (FPS) for YOLOv7, YOLOv8, and YOLOv9. As shown, YOLOv9 not only achieves higher detection accuracy (Precision: 0.950, Recall: 0.946, F1-Score: 0.949) but also benefits from a leaner architecture and faster inference, thereby justifying its optimal selection for our aerial vehicle detection framework.

Table 2: Comparative analysis of YOLOv7, YOLOv8, and YOLOv9 in detection accuracy, complexity, and inference speed

Model	Precision	Recall	F1-score	Parameters (M)	FLOPs (B)	FPS
YOLOv7	0.930	0.920	0.925	37.2	105.2	65
YOLOv8	0.945	0.935	0.937	36.2	93.1	71
YOLOv9	0.950	0.946	0.949	34.9	91.0	76

3.4 Feature Extraction for Enhanced Classification

After YOLOv9 detects vehicles, additional feature extraction techniques refine classification accuracy. While YOLOv9 localizes objects and predicts classes, its large-scale training may miss fine-grained distinctions in aerial imagery. To address this, we integrate Maximally Stable Extremal Regions (MSER), Binary Robust Independent Elementary Features (BRIEF), and Affine-SIFT (ASIFT):

1. MSER enhances edge and contour detection, improving classification under varying lighting.
2. BRIEF provides a compact, efficient descriptor for key vehicle traits with low computational cost.
3. ASIFT ensures scale and viewpoint invariance, maintaining recognition from different angles.

These methods optimize YOLOv9's performance, enhancing classification accuracy, particularly for distinguishing between similar vehicle types.

3.4.1 Rationale for Combining YOLOv9 with Classical Descriptors

YOLOv9 excels in rapid detection but struggles with UAV-specific challenges like scale variations, occlusions, and viewpoint changes. To address this, classical descriptors complement YOLOv9: ASIFT models affine distortions due to UAV altitude, MSER improves localization for small or occluded vehicles, and BRIEF enhances texture and edge pattern recognition. Integrating these descriptors with YOLOv9 reduces overfitting on smaller datasets (e.g., AU-AIR), balancing efficiency and precision while tackling aerial-specific challenges.

3.4.2 Maximally Stable Extremal Regions (MSER) Feature Extraction

Maximally Stable Extremal Regions (MSER) is used for vehicle feature extraction in aerial images, handling size, brightness, and affine variations effectively [25]. MSER identifies stable regions with uniform pixel intensities across different thresholds, making them reliable for vehicle detection in challenging environments, as shown in Eq. (6).

$$R_T = \{x \in \Omega | I(x) \leq T(\text{bright regions})\} \text{ or } R_T = \{x \in \Omega | I(x) \geq T\} (\text{dark regions}) \quad (6)$$

where $\Omega \rightarrow R$ be the intensity function of an image, where $x \in \Omega$ represents the pixel locations. The stability of the region is measured as defined in Eq. (7).

$$\Delta(R_T) = \frac{|R_{T+\delta} - R_T|}{|R_T|} \quad (7)$$

where δ is a small change in the threshold T , and $|R_T|$ is the size of the region. A region is considered maximally stable if $\Delta(R_T)$ reaches a local minimum as mentioned in Eq. (8).

$$\frac{d}{dT} \left(\frac{|R_{T+\delta} - R_T|}{|R_T|} \right) = 0 \quad (8)$$

This equation assures that the regions selected are stable throughout a variety of intensity levels, enhancing the feature extraction process for vehicle recognition in aerial images. The results of MSER can be seen in Fig. 5.



Figure 5: Feature extraction via MSER

3.4.3 Binary Robust Independent Elementary Features (BRIF) Feature Extraction

The feature extraction uses Binary Robust Independent Elementary Features (BRIF) for efficient aerial vehicle detection. Its simplicity and speed make it suitable for large-scale image processing and real-time surveillance. BRIF encodes image patches into binary strings, improving vehicle differentiation and robustness to noise and lighting variations. Descriptors are generated by pixel intensity comparisons, with the formula for point pairs (p_a, p_b) given in Eq. (9).

$$\tau(p; p_a; p_b) = \begin{cases} 1 & \text{if } I(p_a) < I(p_b) \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

The final BRIF descriptor is a binary string formed by concatenating n such comparisons as mentioned in Eq. (10). The results of the BRIF algorithm are illustrated in Fig. 6.

$$d(p) = [\tau(p; p_{a1}; p_{b1}), \tau(p; p_{a2}; p_{b2}), \dots, \tau(p; p_{an}; p_{bn})] \quad (10)$$



Figure 6: Feature extraction via BRIF

3.4.4 Affine-SIFT (ASIFT) Feature Extraction

ASIFT enhances vehicle detection by handling large viewpoint variations in aerial imagery. It extends SIFT's invariance to affine transformations, ensuring stable keypoint extraction across diverse perspectives. ASIFT simulates multiple viewpoints, and SIFT captures invariant keypoints for reliable detection, as defined in Eq. (11).

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (11)$$

where $G(x, y, \sigma)$ is a Gaussian kernel applied to the image $I(x, y)$ at scale σ , and k is a constant multiplicative factor. The keypoints are then described using local gradients, with each keypoint K characterized by a vector $d(K)$ of gradient magnitudes and orientations as mentioned in Eq. (12).

$$d(K) = \sum_{x,y} |\nabla I(x, y)| \cdot e^{-((x-x_0)^2 + (y-y_0)^2)/2\sigma^2} \quad (12)$$

The descriptor vector $d(K)$ for each keypoint is used for matching keypoints between images, while ASIFT ensures that the descriptors remain invariant under affine transformations, making it particularly suited for aerial imagery analysis. The output of the ASIFT algorithm can be seen in Fig. 7.



Figure 7: ASIFT-based feature extraction

3.5 Feature Optimization via QDA

For feature optimization, Quadratic Discriminant Analysis (QDA) was used to enhance the accuracy and robustness of extracted features. QDA models each class as a multivariate Gaussian with distinct covariance matrices, effectively handling non-linear decision boundaries. It optimizes features from MSER, BRIEF, and ASIFT by projecting them into a higher-dimensional space, where posterior probabilities are calculated for precise vehicle classification [26]. Mathematically, QDA classifies a feature vector x by maximizing the posterior probability $P(C_k|x)$, as expressed in Eq. (13).

$$P(C_k|x) \propto \frac{1}{|\sum k|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x - \mu_k)^T \sum_k^{-1} (x - \mu_k)\right) \quad (13)$$

where μ_k represents the mean vector of class C_k , and Σ_k is the covariance matrix of the corresponding class. The optimization is performed by finding the decision rule that minimizes the classification error as defined in Eq. (14).

$$\delta_k(x) = -\frac{1}{2} \log|\sum K| - \frac{1}{2} (x - \mu_k)^T \sum_k^{-1} (x - \mu_k) + \log P(C_k) \quad (14)$$

This optimization process ensures that the feature set is refined to improve classification accuracy in the subsequent module, which involves classifying vehicles using a Probabilistic Neural Network. By leveraging QDA, we achieve a more effective feature representation, reducing the risk of overfitting and improving generalization to unseen data. The results of the QDA are illustrated on Fig. 8.

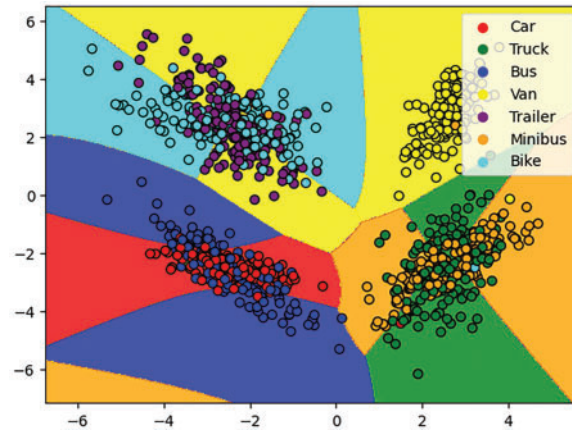


Figure 8: Feature optimization across several classes using QDA

3.6 Classification via Probabilistic Neural Network (PNN)

The final module classifies optimized features using PNN, an efficient classifier for pattern recognition tasks. It estimates class probability densities non-parametrically, leveraging Bayesian decision theory for robust vehicle classification in aerial imagery. PNN processes high-dimensional features optimized by QDA, computing class likelihoods via a kernel-based estimator [27]. Its four-layer architecture input, pattern, summation, and output systematically evaluate and assigns the most probable vehicle class. The posterior probability $P(C_k|x)$ is computed as defined in Eq. (15).

$$P(C_k|x) = \frac{\sum_{i=1}^{n_k} \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right)}{\sum_{j=1}^K \sum_{i=1}^{n_j} \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right)} \quad (15)$$

where x_i is a training sample, σ is the smoothing parameter (kernel width), and K is the number of classes. The PNN classifier computes a weighted sum of these probabilities and assigns the class with the highest probability to the input feature vector. The accuracy of classification is highly dependent on the selection of σ , and tuning this parameter is essential to balance bias and variance in the model. Algorithm 1 shows the workflow for PNN.

Algorithm 1: Harnessing the strength of PNN

Input: Image frames $I = \{image_1, image_2, \dots, image_n\}$

Output: Classification $C = (n_0, n_1, \dots, n_N)$

Detections $D \leftarrow []$: vehicle detections

Feature vector $F \leftarrow []$

Method:

Video = *Video_reader* (V_{path})

Img_frame = read(video)

For $k = 1$ to size (*image frame*)

resize_image = imresize (*image frame* _{k} , 640×640)

(Continued)

Algorithm 1 (continued)

```

segimage = CRF (resizeimage)
D ← YOLOv9 (segmentedimage)
for detection = 1 to size(D)
  F ← MSER (Ddetection)
  F ← BRIEF (Ddetection)
  F ← Affine-SIFT (Ddetection)
  Vehicleclass = PNN (F)
End For
Return Vehicleclass
return imgframe

```

4 Experimentation and Results

The proposed method was implemented and validated in the Python 3.8 environment, utilizing widely used deep learning and image processing libraries. Key dependencies include:

- PyTorch 1.10 (for YOLOv9-based vehicle detection)
- OpenCV 4.5 (for image preprocessing and feature extraction)
- scikit-learn 0.24 (for QDA and PNN classification)
- pydensecrf 1.0 (for CRF-based segmentation)

Experiments were performed on an Intel Core i5-12500H 2.50 GHz processor with 24 GB RAM and an RTX-3050 GPU with 4 GB RAM. The model demonstrated superior performance across both datasets: Roundabout Aerial Image and AU-AIR datasets. The details of the datasets are as follows.

4.1 Dataset Description**4.1.1 Roundabout Aerial Image Dataset**

The Roundabout Aerial Image Dataset comprises 15,474 high-resolution (1920×1080 px) drone-captured images from 8 roundabouts with varying traffic flows. It includes 985,260 instances: 236,850 vehicles, 4899 motorcycles, 2262 trucks, 1752 buses, and 552 empty roundabouts. An additional 46,422 images were generated via data augmentation. Annotated for vehicle detection and classification across four classes, this dataset supports research in object detection and trajectory analysis in complex traffic scenarios.

4.1.2 AU-AIR Dataset

The AU-AIR dataset, designed for low-altitude UAV-based detection, contains 32,823 labeled instances across seven vehicle categories: Car, Truck, Bus, Cycle, Van, Trailer, and Bike. It introduces challenges like occlusions, motion blur, and complex backgrounds, reflecting real-world aerial surveillance. The class distribution includes Cars (24,581), Trucks (3102), Buses (1854), Cycles (2412), Vans (874), Trailers (650), and Bikes (350).

4.2 Model Evaluation and Experimental Results

We evaluated the system on three datasets, repeating trials five times for accuracy. Table 3 shows the precision, recall, and F1-score of the detection algorithm, while Table 4 compares YOLOv9 with the proposed model (YOLOv9 + QDA + PNN). Results confirm that QDA optimization and PNN classification enhance performance, with variations due to dataset characteristics.

Table 3: Precision, recall, and F1-score for the detection algorithm

Datasets	Precision	Recall	F1-score	No of classes
Roundabout	0.9502	0.9460	0.9487	4
AU-AIR	0.9437	0.9422	0.9428	7

Table 4: YOLOv9 vs. proposed model performance across both datasets

Datasets	Model	Precision	Recall	F1-score
Roundabout	YOLOv9 (Detection Only)	0.9502	0.9460	0.9487
	YOLOv9 + QDA + PNN (Proposed Model)	0.9589	0.9505	0.9547
AU-AIR	YOLOv9 (Detection Only)	0.9437	0.9422	0.9428
	YOLOv9 + QDA + PNN (Proposed Model)	0.9513	0.9440	0.9476

Tables 5 and 6 show vehicle detection metrics for the Roundabout and AU-AIR datasets, with Figs. 9 and 10 displaying the classification confusion matrices. Tables 7 and 8 compare classification performance, while Tables 9 and 10 benchmark against state-of-the-art models. Table 11 analyzes computational complexity.

Table 5: Vehicle detection having precision, recall, and F1-score evaluation of the Roundabout Aerial Image dataset

Classes	Precision (D)	Recall (D)	F1-score (D)
Car	0.9500	0.9500	0.9500
Cycle	0.9485	0.9402	0.9447
Truck	0.9501	0.9600	0.9552
Bus	0.9523	0.9341	0.9451
Mean	0.9502	0.9460	0.9487

Table 6: Vehicle detection having precision, recall, and F1-score evaluation of the AU-AIR dataset

Classes	Precision (D)	Recall (D)	F1-score (D)
Car	0.9536	0.9745	0.9610
Truck	0.9478	0.9180	0.9382
Bus	0.9491	0.9421	0.9493
Cycle	0.9249	0.9429	0.9352
Van	0.9157	0.9195	0.9109
Trailer	0.9733	0.9631	0.9604
Bike	0.9417	0.9355	0.9447
Mean	0.9437	0.9422	0.9428

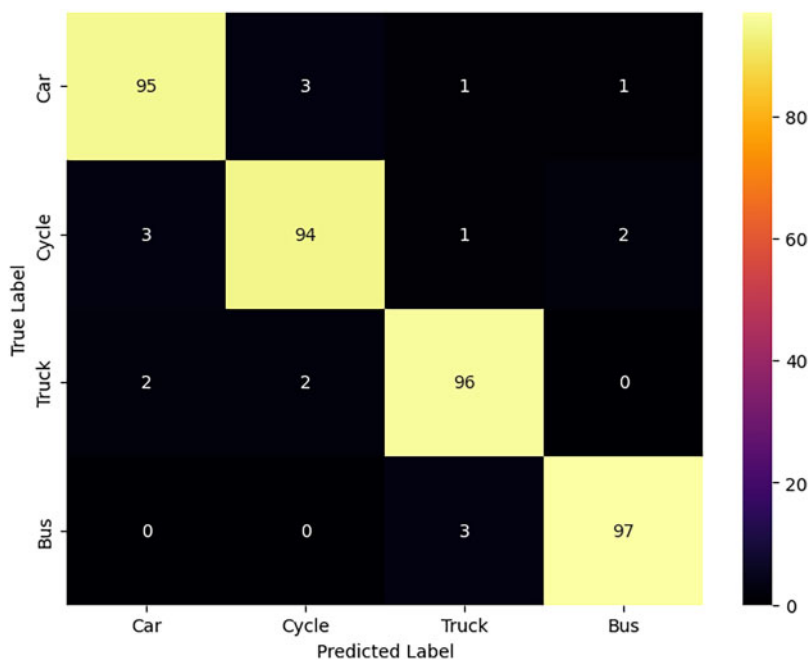


Figure 9: Confusion matrix for individual class accuracy over the Roundabout Aerial Image dataset

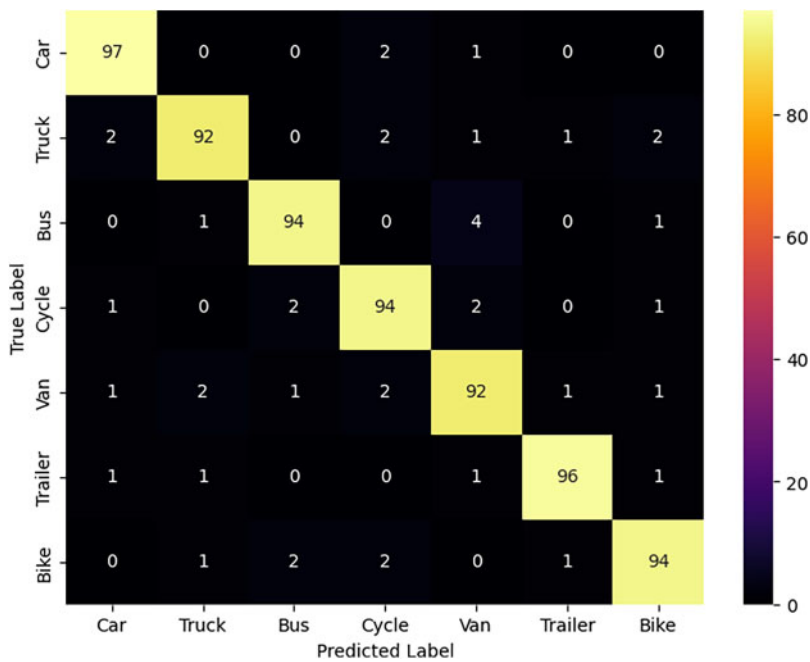


Figure 10: Confusion matrix for individual class accuracies over the AU-AIR dataset

Table 7: Vehicle classification having precision, recall, and F1-score evaluation of the Roundabout dataset

Classes	Precision (C)	Recall (C)	F1-score (C)
Car	0.9500	0.9500	0.9500
Cycle	0.9490	0.9400	0.9445
Truck	0.9600	0.9600	0.9600
Bus	0.9796	0.9700	0.9748

Table 8: Vehicle classification having precision, recall, and F1-score evaluation of the AU-AIR dataset

Classes	Precision (C)	Recall (C)	F1-score (C)
Car	0.9700	0.9700	0.9700
Truck	0.9592	0.9203	0.9392
Bus	0.9400	0.9380	0.9400
Cycle	0.9400	0.9400	0.9400
Van	0.9394	0.9200	0.9296
Trailer	0.9600	0.9600	0.9600
Bike	0.9500	0.9400	0.9450

Table 9: Comparison of model detection rate with other state-of-the-art methods

Datasets	Models	Precision
Roundabout	Single shot detector (SSD) [28]	0.80
	RetinaNet [29]	0.86
	Faster RCNN [30]	0.82
	Proposed method	0.95
AU-AIR	Yolov4 [31]	0.81
	Yolov3 [31]	0.69
	Blob detection [32]	0.87
	Proposed method	0.943

Table 10: Classification Comparison with other state-of-the-art models

Method	Roundabout	AU-AIR
Wang et al. [33]	91.40%	–
Mujtaba and Jalal [34]	91%	–
Bozcan and Kayacan [35]	–	88.2%
Yusuf et al. [36]	–	87.0%
Qureshi et al. [37]	–	93.1%
Proposed method	95.54%	94.14%

Table 11: Computational complexity analysis

Methods	Computational complexity	Execution time (s)	Optimized time
Pre-processing	$O(m \cdot n)$, $\Theta(m \cdot n)$, $\Omega(m)$ m = pixels, n = images	0.40	0.40
Segmentation (CRF)	$O(m \cdot \log n)$ $\Theta(m \cdot \log n)$ $\Omega(m)$, m = pixels, n = segments	0.60	0.20
Vehicle detection (YOLOv9)	$O(n^2)$, $\Theta(n \cdot p)$, $\Omega(n)$, n = images, p = features	0.75	0.52
Feature extraction (MSER, BRIEF, Affine-SIFT)	$O(n \cdot m)$, $\Theta(n \cdot m)$, $\Omega(n)$, n = features, m = images	0.88	0.53
Feature optimization (QDA)	$O(n^3)$, $\Theta(n^2)$, $\Omega(n)$, n = vehicles	0.50	0.25
Classification (PNN)	$O(n^2)$, $\Theta(n \cdot p)$, $\Omega(n)$	0.85	0.30

The confusion matrix demonstrates strong classification across Car, Cycle, Truck, and Bus classes, with recall rates exceeding 94%. Minor misclassifications between Car–Cycle and Truck–Bus likely result from structural similarities, occlusions, or dataset bias. While the model shows robust performance, further refinement is needed to better distinguish similar vehicle types.

The confusion matrix shows strong classification across all seven classes, with Car, Bus, and Bike achieving over 94% recall. Minor confusions occur between Truck–Van, Van–Trailer, and Cycle–Bike/Bus, likely due to shape similarities and perspective distortions. Despite these, the model performs excellently, with future work focused on enhancing feature extraction for better class separability.

Table 11 analyzes computational complexity, including worst-case, average-case, and best-case complexities. Image preprocessing and segmentation are logarithmic, YOLOv9 detection is quadratic, feature extraction and QDA are polynomial, and PNN classification is quadratic in high-dimensional spaces. Execution times balance accuracy and efficiency for real-time detection. Table 12 presents the ablation study across both datasets.

Table 12: Ablation study evaluating the impact of different components in our proposed model across the Roundabout Aerial image and AU-AIR datasets

Experiment	Radiometric calibration	CRF segmentation	YOLOv9 detection	MSER	BRIEF	ASIFT	QDA optimization	PNN	Roundabout (%)	AU-AIR (%)
Full model (Baseline)	✓	✓	✓	✓	✓	✓	✓	✓	95.5	94.1
Without radiometric calibration	✗	✓	✓	✓	✓	✓	✓	✓	92.3	91.5
Without CRF	✓	✗	✓	✓	✓	✓	✓	✓	91.8	90.9
Without YOLOv9 detection	✓	✓	✗	✓	✓	✓	✓	✓	88.7	87.2
Without MSER	✓	✓	✓	✗	✓	✓	✓	✓	93.2	92.1
Without BRIEF	✓	✓	✓	✓	✗	✓	✓	✓	94.0	93.2
Without ASIFT	✓	✓	✓	✓	✓	✗	✓	✓	92.8	91.6
Without QDA optimization	✓	✓	✓	✓	✓	✓	✗	✓	90.5	89.4

(Continued)

Table 12 (continued)

Experiment	Radiometric calibration	CRF segmentation	YOLOv9 detection	MSER	BRIEF	ASIFT	QDA optimization	PNN	Roundabout (%)	AUAIR (%)
With simple classifier	✓	✓	✓	✓	✓	✓	✓	SVM	89.7	88.2
YOLOv9 + ResNet-50	✓	✓	✓	✗	✗	✗	✗	ResNet-50	91.3	89.7
YOLOv9 + inception-v3	✓	✓	✓	✗	✗	✗	✗	Inception	92.1	90.2
YOLOv9 + ViT-base	✓	✓	✓	✗	✗	✗	✗	ViT	93.5	91.8

The ablation study confirms that the full model achieves the highest accuracy, with each module essential. Removing any module, including Radiometric Calibration, CRF, YOLOv9, MSER, BRIEF, ASIFT, or QDA, reduces performance, while replacing PNN with a basic classifier weakens accuracy. Table 11 shows that the proposed model outperforms CNN/ViT classifiers by 2–4%, highlighting the importance of hybrid features.

5 Discussion/Limitations

The framework shows strong vehicle detection in urban daytime scenarios but has limitations. Evaluation is limited to the Roundabout Aerial Image and AU-AIR datasets, lacking geographic and temporal diversity. MSER and BRIEF may underperform in low-contrast or noisy environments. The framework assumes fixed UAV altitude and sensor configurations, affecting performance with variations. Class imbalance in AU-AIR introduces bias, limiting generalizability. Current gaps highlight the need for multi-modal datasets, adaptive extraction, and ethical focus. The YOLOv9 + QDA + PNN model improves vehicle classification, surpassing existing methods in precision, recall, and F1-score. While QDA optimizes features and PNN enhances class separability, limitations include the absence of pedestrian detection and higher computational cost. Fig. 11 illustrates system limitations.



Figure 11: Limitation results of the proposed system

5.1 Scalability and Real-Time Feasibility

To reconcile accuracy with real-world operational constraints, we propose the following optimizations:

1. **Model Pruning & Quantization:** Reducing parameters in YOLOv9 with pruning and INT8 quantization cuts model size and inference time by ~30% with minimal accuracy loss.
2. **Feature Selection Optimization:** Using mutual information scoring, prioritizing ASIFT for viewpoint invariance, reducing extraction time by ~25%.
3. **Efficient Classification:** Replacing PNN with MobileNetV3 or pruned ResNet reduces classification time by 65% while maintaining >92% accuracy. Using PCA/t-SNE instead of QDA simplifies optimization.

4. **Parallel Processing & Edge Computing:** Using UAV-compatible hardware and hybrid cloud-UAV frameworks achieves 2–3× speedups, improving runtime to 1.20 s/image with 93.8% accuracy on Roundabout and 92.6% on AU-AIR, a 70% improvement.

This optimization achieves real-time performance with slight accuracy trade-offs, ideal for UAV-based traffic monitoring. Further speedup is possible through hardware-software co-design or hybrid frameworks.

5.2 Deployment Considerations: Ground Station vs. Onboard Inference

The proposed method is implemented as a post-processing pipeline on a ground station with sufficient computational resources, enabling full use of the multi-stage framework. Although designed for off-board processing, [Section 5.1](#) discusses optimizations for real-time onboard UAV inference, including model pruning, INT8 quantization, and feature selection to reduce complexity and latency.

5.3 Root Causes of Class-Wise Performance Disparities in Aerial Vehicle Detection

The disparities in precision and recall across vehicle classes can be attributed to the following factors:

- Trucks (AU-AIR): Lower recall due to underrepresentation, size, and occlusion. Feature ambiguity with buses and trailers also causes misclassification.
- Cycles and Bikes: Lower precision due to small size and motion blur, affecting descriptor reliability.
- Buses (Roundabout): Higher precision due to distinctive shape and consistent placement in roundabouts.

These variations underscore the impact of class imbalance, scale variability, and occlusion frequency on model performance.

6 Conclusion

This work presents a multi-stage vehicle detection and classification framework using YOLOv9 for detection and Probabilistic Neural Network (PNN) for classification in aerial imagery. Feature extraction techniques like MSER, BRIEF, and Affine-SIFT enhance classification accuracy. Experiments on Roundabout Aerial Image and AU-AIR datasets show YOLOv9 achieving high detection precision (0.9502 for Roundabout, 0.942 for AU-AIR), while PNN attains 95.54% and 94.14% accuracy, respectively. Our approach outperforms state-of-the-art methods in precision, recall, and F1-score. Future work can integrate pedestrian recognition and optimize computational efficiency for real-time applications.

Acknowledgement: The authors are thankful Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R508), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Funding Statement: This work was supported through Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R508), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The research team thanks the Deanship of Graduate Studies and Scientific Research at Najran University for supporting the research project through the Nama'a program, with the project code NU/GP/SERC/13/18-5.

Author Contributions: Study conception and design: Abdulwahab Alazeb; data collection: Muhammad Hanzla and Naif Al Mudawi; analysis and interpretation of results: Mohammed Alshehri and Haifa F. Alhasson; draft revision: Dina Abdulaziz AlHammadi; draft manuscript preparation: Ahmad Jalal. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All publicly available datasets are used in the study.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Elmeseiry N, Alshaer N, Ismail T. A detailed survey and future directions of unmanned aerial vehicles (UAVs) with potential applications. *Aerospace*. 2021;8(12):363. doi:10.3390/aerospace8120363.
2. Videras Rodríguez M, Melgar SG, Cordero AS, Márquez JMA. A critical review of unmanned aerial vehicles (UAVs) use in architecture and urbanism: scientometric and bibliometric analysis. *Appl Sci*. 2021;11(21):9966. doi:10.3390/app11219966.
3. Li X, Savkin AV. Networked unmanned aerial vehicles for surveillance and monitoring: a survey. *Future Internet*. 2021;13(7):174. doi:10.3390/fi13070174.
4. Maity M, Banerjee S, Chaudhuri SS. Faster R-CNN and YOLO-based vehicle detection: a survey. In: *Proceedings of the 2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*; 2021 May 6; Erode, India. p. 1442–7. doi:10.1109/ICCMC51019.2021.9418274.
5. Bathla G, Bhadane K, Singh RK, Kumar R, Aluvalu R, Krishnamurthi R, et al. Autonomous vehicles and intelligent automation: applications, challenges, and opportunities. *Mob Inf Syst*. 2022;2022(1):7632892. doi:10.1155/2022/7632892.
6. Bakirci M. Utilizing YOLOv8 for enhanced traffic monitoring in intelligent transportation systems (ITS) applications. *Digit Signal Process*. 2024;152(2):104594. doi:10.1016/j.dsp.2024.104594.
7. Bakirci M. Real-time vehicle detection using YOLOv8-nano for intelligent transportation systems. *Trait Du Signal*. 2024;41(4):1727–40. doi:10.18280/ts.410407.
8. Liu C, Ding Y, Zhu M, Xiu J, Li M, Li Q. Vehicle detection in aerial images using a fast oriented region search and the vector of locally aggregated descriptors. *Sensors*. 2019;19(15):3294. doi:10.3390/s19153294.
9. Hamadi R, Ghazzai H, Massoud Y. Image-based automated framework for detecting and classifying unmanned aerial vehicles. In: *Proceedings of the 2023 IEEE International Conference on Smart Mobility (SM)*; 2023 Mar 19–21; Thuwal, Saudi Arabia. p. 149–53. doi:10.1109/SM57895.2023.10112531.
10. Arinaldi A, Pradana JA, Gurusinga AA. Detection and classification of vehicles for traffic video analytics. *Procedia Comput Sci*. 2018;144:259–68. doi:10.1016/j.procs.2018.10.527.
11. Kumar S, Jain A, Rani S, Alshazly H, Idris SA. Deep neural network-based vehicle detection and classification of aerial images. *Intell Autom Soft Comput*. 2022;34(1):117–30. doi:10.32604/iasc.2022.024812.
12. Goecks VG, Woods G, Valasek J. Combining visible and infrared spectrum imagery using machine learning for small unmanned aerial system detection. *arXiv:2003.12638*. 2020. doi:10.1117/12.2557442.
13. Lee EJ. Validation of object detection in UAV-based images using synthetic data. *arXiv:2201.06629*. 2022. doi:10.1117/12.2586860.
14. Li W, Li H, Wu Q, Chen X, Ngan KN. Simultaneously detecting and counting dense vehicles from drone images. *IEEE Trans Ind Electron*. 2019;66(12):9651–62. doi:10.1109/TIE.2019.2899548.
15. Isaac-Medina BK, Poyser M, Organisciak D, Willcocks CG, Breckon TP, Shum HP. Unmanned aerial vehicle visual detection and tracking using deep neural networks: a performance benchmark. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*; 2021 Oct 11–17; Montreal, BC, Canada. p. 1223–32. doi:10.1109/ICCVW54120.2021.00142.
16. Deng Z. Toward fast and accurate vehicle detection in aerial images using coupled region-based convolutional neural networks. *IEEE J Sel Top Appl Earth Obs Remote Sens*. 2017;10(8):3652–64. doi:10.1109/JSTARS.2017.2694890.
17. Tan Y. Vehicle detection and classification in aerial imagery. In: *Proceedings of the IEEE International Conference on Image Processing (ICIP)*; 2018 Oct 7–10; Athens, Greece. p. 86–90. doi:10.1109/ICIP.2018.8451709.
18. Jiang L, Zhang Z, Hu Y. Unmanned aerial vehicle hyperspectral remote sensing image classification based on parallel convolutional neural network. In: *Proceedings of the International Conference on Information Systems and Computing Technology (ISCTech)*; 2023 Jul 30–Aug 1; Qingdao, China. p. 526–30. doi:10.1109/ISCTech60480.2023.00099.

19. Zhou Jie, Xu H, Zhou R, Du X. Based on improved lightweight YOLOv8 for vehicle detection. *Adv Comput Mater Sci Res.* 2024;1(1):293–300. doi:10.70114/acmsr.2024.1.1.P293.
20. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-end object detection with transformers. In: *Proceedings of the European Conference on Computer Vision*; 2020 Aug 23–28; Glasgow, UK. Berlin/Heidelberg, Germany: Springer. p. 213–29. doi:10.1007/978-3-030-58452-8_13.
21. Tan M, Pang R, Le QV. EfficientDet: scalable and efficient object detection. *arXiv:1911.09070*. 2020. doi:10.48550/arXiv.1911.09070.
22. Daniels L, Eeckhout E, Wieme J, Dejaegher Y, Audenaert K, Maes WH. Identifying the optimal radiometric calibration method for UAV-based multispectral imaging. *Remote Sens.* 2023;15(11):2909. doi:10.3390/rs15112909.
23. Jiang X, Yu H, Lv S. An image segmentation algorithm based on a local region conditional random field model. *Int J Commun Netw Syst Sci.* 2020;13(9):139–59. doi:10.4236/ijcns.2020.139009.
24. Lv C, Mittal U, Madaan V, Agrawal P. Vehicle detection and classification using an ensemble of EfficientDet and YOLOv8. *PeerJ Comput Sci.* 2024;10(19):e2233. doi:10.7717/peerj-cs.2233.
25. Ramya PP, Ajay J. Object recognition and classification based on improved bag of features using SURF and MSER local feature extraction. In: *Proceedings of the International Conference on Innovations in Information and Communication Technology (ICIICT)*; 2019 Apr 25–26; Chennai, India. p. 1–4. doi:10.1109/ICIICT1.2019.8741434.
26. Ghosh A, SahaRay R, Chakrabarty S, Bhadra S. Robust generalized quadratic discriminant analysis. *Pattern Recognit.* 2021;117(6):107981. doi:10.1016/j.patcog.2021.107981.
27. Hashima S, Saad MH, Hatano K, Rizk H. Vehicle classification in intelligent transportation systems using deep learning and seismic data. In: *Proceedings of the IEEE International Conference on Intelligence and Security Informatics (ISI)*; 2023 Oct 2–3; Charlotte, NC, USA. p. 1–6. doi:10.1109/ISI58743.2023.10297252.
28. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. SSD: single shot multibox detector. In: *Proceedings of the Computer Vision-ECCV 2016: 14th European Conference*; 2016 Oct 11–14; Amsterdam, The Netherlands. p. 21–37. doi:10.1007/978-3-319-46448-0_2.
29. Lin TY, Goyal P, Girshick R, He K, Dollar P. Focal loss for dense object detection. *IEEE Trans Pattern Anal Mach Intell.* 2017;42(2):318–27. doi:10.1109/TPAMI.2018.2858826.
30. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell.* 2017;39(6):1137–49. doi:10.1109/TPAMI.2016.2577031.
31. Xu H, Cao Y, Lu Q, Yang Q. Performance comparison of small object detection algorithms of UAV-based aerial images. In: *Proceedings of the 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*; 2020 Oct 16–19; Hong Kong, China. p. 16–9. doi:10.1109/DCABES50732.2020.00014.
32. Qureshi AM, Almujaally NA, Alotaibi SS, Alatiyyah MH, Park J. Intelligent traffic surveillance through multi-label semantic segmentation and filter-based tracking. *Comput Mater Contin.* 2023;76(3):3707–25. doi:10.32604/cmc.2023.040738.
33. Wang CY, Bochkovskiy A, Liao HM. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; 2023 Jun 17–24; Vancouver, BC, Canada. p. 7464–75. doi:10.1109/CVPR52729.2023.00721.
34. Mujtaba G, Jalal A. Remote sensing-based vehicle monitoring system using YOLOv10 and CrossViT. In: *2024 26th International Multi-Topic Conference (INMIC)*; 2024; Karachi, Pakistan. p. 1–6. doi:10.1109/INMIC64792.2024.11004315.
35. Bozcan I, Kayacan E. AU-AIR: a multi-modal unmanned aerial vehicle dataset for low altitude traffic surveillance. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*; 2020 May 1–Aug 31; Paris, France. p. 8504–10. doi:10.1109/ICRA40945.2020.9196845.
36. Yusuf MO, Hanzla M, Rahman H, Sadiq T, Al Mudawi N, Almujaally NA, et al. Enhancing vehicle detection and tracking in UAV imagery: a pixel labeling and particle filter approach. *IEEE Access.* 2024;12(11):72896–911. doi:10.1109/ACCESS.2024.3401253.
37. Qureshi AM, Algarni A, Aljuaid H. Semantic segmentation based real-time traffic monitoring via Res-UNet classifier and kalman filter. *SN Comput Sci.* 2025;6(1):30. doi:10.1007/s42979-024-03586-7.