Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: jeeemi.org; Vol. 7, No. 4, October 2025, pp: 1240-1258 e-ISSN: 2656-8632

RESEARCH ARTICLE OPEN ACCESS

Automatic Target Recognition using Unmanned Aerial Vehicle Images with Proposed YOLOv8-SR and Enhanced Deep Super-Resolution Network

Gangeshwar Mishra[1], Rohit Tanwar[2] and Prinima Gupta[1]

[1] Department of CST, Manav Rachna University, Faridabad, Haryana, India.
[2] School of CSE, Shri Mata Vaishno Devi University, Katra, Jammu & Kashmir, India.

Corresponding author: Rohit Tanwar (e-mail: rohit.tanwar.cse@gmail.com), **Author(s) Email**: Gangeshwar Mishra (e-mail: gangeshwarmishra045@gmail.com, Prinima Gupta (e-mail: prinima@mru.edu.in)

Abstract **Modern surveillance necessitates the use of automatic target recognition (ATR) to identify targets or objects quickly and accurately for multiclass classification in unmanned aerial vehicles (UAVs) such as pedestrians, people, bicycles, cars, vans, trucks, tricycles, buses, and motors. The inadequate recognition rate in target detection for UAVs could be due to the fundamental issues provided by the poor resolution of photos recorded from the distinct perspective of the UAVs. The VisDrone dataset used for image analysis consists of a total of 10,209 UAV photos. This research work presents a comprehensive framework specifically for multiclass target classification using VisDrone UAV imagery. The YOLOv8-SR, which stands for "You Only Looked Once Version 8 with Super-Resolution," is a developed model that builds on the YOLOv8s model with the Enhanced Deep Super-Resolution Network (EDSR). The YOLOv8-SR uses the EDSR to convert the low-resolution image to a high-resolution image, allowing it to estimate pixel values for better processing better. The high-resolution image was generated by the EDSR model, having a Peak Signal-to-Noise Ratio (PSNR) of 25.32 and a Structural Similarity Index (SSIM) of 0.781. The YOLOv8-SR model's precision is 63.44%, recall is 46.64%, F1-score is 52.69%, mean average precision (mAP@50) is 51.58%, and the mAP@50–95 is 50.67% over the range of confidence thresholds. The investigation fundamentally transforms the precision and effectiveness of ATR, indicating a future in which ingenuity overcomes obstacles that were once considered insurmountable. This development is characterized by the use of an improved deep super-resolution network to produce super-resolution images from low-resolution inputs. The YoLov8-SR model, a sophisticated version of the YoLov8s framework, is key to this breakthrough. By amalgamating the EDSR methodology with the advanced YOLOv8-SR framework, the system generates high-resolution images abundant in detail, markedly exceeding the informational quality of their low-resolution versions.**

Keywords **Deep Learning, High Resolution, Image Processing, Object Detection, YOLOv8**.

I. Introduction

The low resolution of pictures taken from the unique perspective of UAVs contributes to the inadequate identification rate in UAV target detection. The aerial photos, which have low resolution and limited pixel data, make it difficult to identify and locate things of interest accurately. The lack of resolution hindered traditional object identification deep learning algorithms, limiting their capacity to distinguish intricate features and accurately identify items [1], [2]. The challenge of reliable object detection in low-resolution UAV imagery necessitates specialized solutions for effective target recognition in complex environments [3]. Deep learning models such as YOLOv8, developed by Ultralytics, address this problem with a single-stage object detection paradigm that enables real-time recognition [4], [5]. YOLOv8 is an advanced iteration of the YOLO series, optimizing both accuracy and inference speed [6]. To further enhance efficiency, YOLOv8s, a smaller variant, reduces network size and parameters, prioritizing resource optimization while maintaining precision [4], [7]. Both YOLOv8 and YOLOv8s operate using a grid-based framework that includes image preprocessing, feature extraction, target detection, and post-processing via non-maximum suppression (NMS) to refine results by eliminating duplicate detections [4], [5]. Anchor boxes

Manuscript Received 05 May 2025; Revised 20 August 2025; Accepted 5 October 2025; Available online 14 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.888

facilitate precise bounding box predictions, with YOLOv8s employing fewer anchor variations to improve computational efficiency [7], [8]. While YOLOv8 emphasizes accuracy, YOLOv8 maximizes speed and resource efficiency, making it particularly suitable for real-time UAV applications that demand rapid inference within constrained processing capabilities [3],[4],[5]. By refining detection mechanisms and optimizing network structure, YOLOv8 models contribute to overcoming the limitations of UAV-based object recognition, advancing autonomous visual processing in aerial contexts.

While YOLOv8s offers speed advantages for UAV deployment, its accuracy is compromised when processing low-resolution UAV imagery. The model's simplification (reduced parameters/anchor boxes) inherently limits its ability to recover fine-grained details crucial for detecting small or indistinct objects in low pixel count aerial images. Existing YOLO variants lack integrated mechanisms to enhance input resolution effectively.

This method integrates the EDSR (Enhanced Deep Super-Resolution) network with the YOLOv8s architecture. EDSR first processes the low-resolution UAV input image to generate a high-resolution counterpart. This super-resolved image is then fed into the YOLOv8s model for object detection. The hybrid design aims to overcome the low-resolution limitation by enhancing input quality before detection, leveraging EDSR's proven ability to recover image detail and YOLOv8's efficient detection capabilities. Simplifying the model improved its performance without affecting its capacity to handle objects of various sizes and forms.

To develop a robust object detection solution (YOLOv8-SR) that overcomes the challenge of low-resolution UAV perspective images by integrating super-resolution through EDSR with the efficient YOLOv8s detector, enabling reliable and high-performing target recognition in aerial applications.

This research advances real-time UAV object detection by optimizing and proposing YOLOv8 models for efficiency and accuracy in a constrained environment. One of the key contributions is introducing a new hybrid architecture, YOLOv8-SR, combining the Enhanced Deep Super-Resolution (EDSR) network with the lightweight YOLOv8s detector. This is the first known integration of such, intended to solve the serious issue of low-resolution UAV images. By improving input resolution with EDSR prior to detection, the model enhances significantly the capacity of YOLOv8s to detect features from low pixel inputs. Without any increase in complexity, the design preserves the computational demands required for real-time UAV usage. Experimental verification verifies that YOLOv8-SR surpasses baseline YOLOv8s and competing approaches in target detection precision at no prohibitive computational cost.

The research paper follows a structured approach to YOLOv8-SR's development and evaluation. Section II. reviews YOLOv8/YOLOv8s, EDSR, and prior UAV research methodology. Section III details about proposed YOLOv8-SR: An Improved YOLOv8s variant for Super-Resolution with EDSR, and its methodology. Section IV presents the datasets, preprocessing, evaluation, and analysis of YOLOv8-SR super-resolution by EDSR comparisons, results visualizations, ablation studies, and comparative analysis of the results. Section V presents a brief discussion on comparison and limitations. Section VI summarizes the research problem, contributions, findings, and future directions.

II. Methodology

A. Image preprocessing

In order to ensure high quality, it was necessary to preprocess each image from the VisDrone dataset. The dataset exhibits a range of image sizes and resolutions, spanning from low to high. Nevertheless, we conducted preprocessing on all photographs by shrinking them to dimensions of 640x640. Utilizing deep learning models such as YOLOv8, the input image can be preprocessed by applying a Gaussian blur and morphological smoothing to remove noise. The initial step involves preparing the data for the YOLOv8, which is necessary for data interpretation and preprocessing [9], [10].

B. YOLOv8s

The YOLOv8s architecture is designed with a depth coefficient of 0.33, a width coefficient of 0.50, a feature size of 1024 bits, and a scaling ratio of 2.0. This setup provides the best balance between detection precision and computational efficiency and is very well-suited for real-time object detection purposes. In comparison to other YOLOv8 variants such as YOLOv8n, YOLOv8m, YOLOv8l, and YOLOv8x, the YOLOv8s model has lowered depth and width parameters, making it smaller in terms of trainable parameters but quicker in inference speed with minimal loss in precision. However, due to its exceptional balance between speed and accuracy, YOLOv8s remains widely used in many applications. The device's small size and effective processing make it a useful alternative for devices with limited resources, where quicker processing is essential for maximum performance [4], [11]. The procedure for target recognition with YOLOv8s encompasses the following steps:

Step 1- Input

The YOLOv8s model receives the input image, which contains some objects for recognition, such as pedestrians, people, bicycles, cars, vans, trucks, tricycles, buses, and motors.

Manuscript Received 05 May 2025; Revised 20 August 2025; Accepted 5 October 2025; Available online 14 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.888

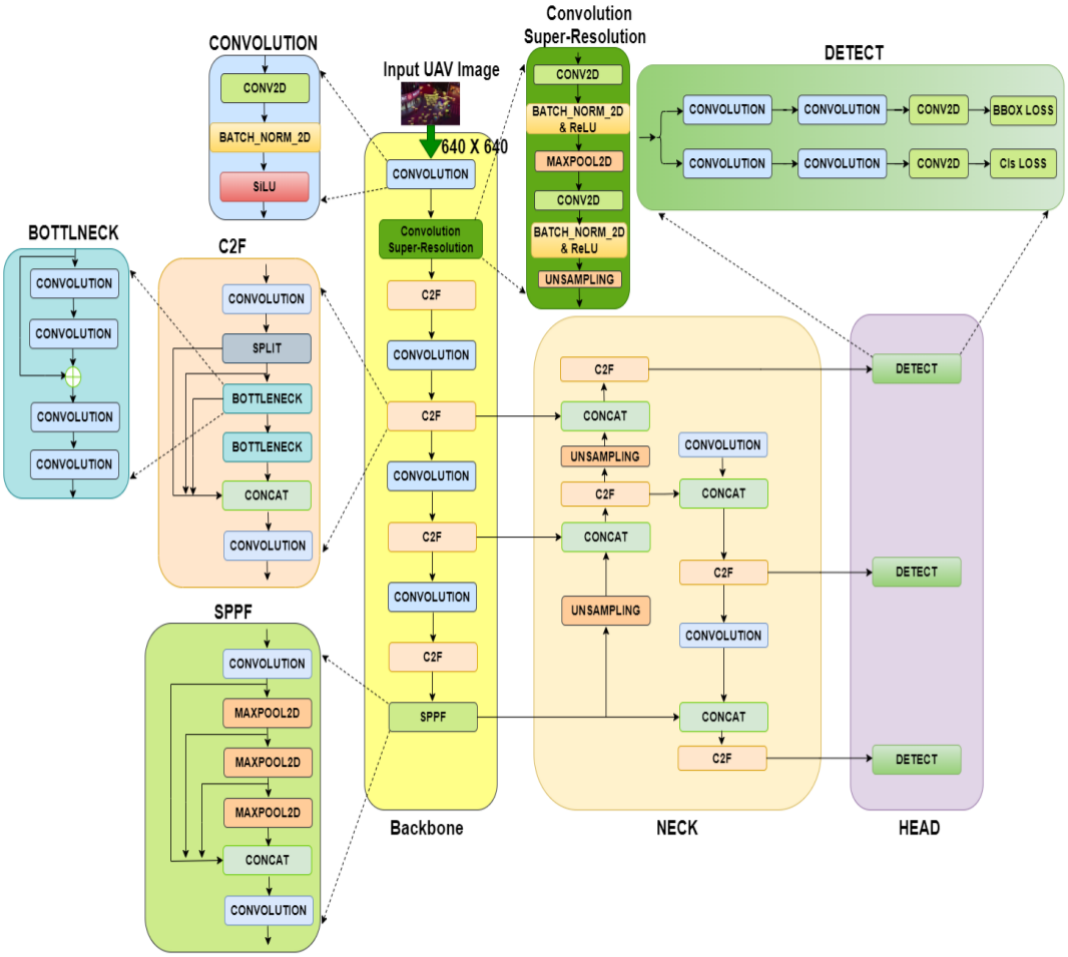

Fig. 1. YOLOv8SR (YOLOv8s detector with integrated EDSR super-resolution backbone for enhanced small-object features.)

Step 2 - Structure

Optimized for computational efficiency, the YOLOv8s architecture excels in real-time object identification across various platforms. The YOLOv8s architecture consists of three fundamental components: the backbone, neck, and head. These components carry out all the necessary computations. Each block's function is outlined below. Fig. 1 depicts the structure that YOLOv8s executes [4], [5].

Step 2.1 - Backbone

The YOLOv8s model utilizes a novel backbone design, also known as the feature extractor, which has a vital function in extracting significant features from the incoming data. During its operation, the backbone performs a variety of actions. This architecture employs a compound scaling technique that optimizes the depth, breadth, and resolution of the network to ensure efficient and effective feature extraction. Initially, it detects fundamental patterns, such as edges and textures, in the very first layers. Furthermore, it can embrace many scales of representation, allowing it to capture features at different levels of abstraction. The backbone ultimately creates a detailed and complex hierarchical representation of the input image, which includes a variety of significant features [5], [7].

Step 2.1.1 - Convolution Block

In YOLOv8s, the convolution block typically consists of convolutional layers, followed by additional processes like batch normalization and activation functions. The precise specifications of the convolution block may vary depending on the variants; however, the following equation provides a basic representation for a single convolutional layer inside the block:

$$Y = SiLU(convolution(X, W) + B) \quad (1)$$

A collection of adaptable convolutional filters, symbolized as W, and an additional term known as the bias, B, subject the input feature map, labeled as X, to a process of alteration as presented in Eq. (1) [12]. The convolution procedure entails the process of sliding these filters across the input, doing element-wise multiplications, and then summing the outcomes. Afterwards, a Sigmoid Linear Units (SiLU) activation

Manuscript Received 05 May 2025; Revised 20 August 2025; Accepted 5 October 2025; Available online 14 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.888

function is applied to each element of the resultant feature map as shown in Eq. (2) [13].

$$SiLU(x) = x.Sigmoid(x) \quad (2)$$

The output feature map, denoted as Y, as presented in Eq.(1), contains the modified information that is prepared for further processing in the network [11] , [14].

Step 2.1.2 – Cross-stage partial bottleneck with two convolutions (C2f)

The C2F module in YOLOv8s serves to link the backbone to the Feature Pyramid Network (FPN), including more skip links and extra split operations. The C2F block is often used to reduce the resolution of the feature maps that are supplied and capture significant features from the backbone. It aids in extracting characteristics at varying spatial resolutions. The equation for the C2F block can be expressed as follows:

$$F_i = conv_{block(F_{\{i-1\}})} \quad (3)$$

Feature map F_i represents the i^{th}stage and F_{i-1} represents the prior i-1th stage. The $conv_{block}$is the convolution block with convolutional layers, as presented in Eq.(3) [7]. As the input travels across the network, this structure gradually extracts hierarchical characteristics, improving model performance and learning [7] , [15].

Step 2.1.3 - SPPF (Spatial Pyramid Pooling and Fusion) Block

The SPPF block is used to collect features at various sizes and combine them into a unified feature map. It enhances the network's capacity to identify objects of varying sizes. In the SPPF block, spatial pyramid pooling is used at different scales, followed by separate convolution blocks. Finally, the feature maps are joined together to make a multi-scale feature representation. The equations governing the SPPF block may be expressed in the following manner:

$$X_i = spp_{conv_{block(X)}} \quad (4)$$

$$Y_i = conv_{block(Y_{\{i-1\}})} \quad (5)$$

$$Y = concatenate([Y_1, Y_2, \dots, Y_n]) \quad (6)$$

The input feature map is X, while the intermediate feature map at the ith scale is Y_i. By using concatenate to combine features from different scales, Y represents the final concatenated feature map. The input feature map is convolutionally processed by the $spp_{conv_{block}}$, which processes each scale. This technique may also enhance features using the $conv_{block}$ discussed before as presented in Eq. (4). This multi-scale technique improves the network's spatial information acquisition, making target identification and semantic segmentation more successful. Eqs. (4), (5), (6) illustrate key computational formulations relevant to the methodology, supporting the theoretical framework and experimental validation presented in [4], [16].

Step 2.2 - Neck

The neck functions as an essential component in establishing a connection between the backbone and the head of a network, facilitating the integration of contextual information and running feature fusion operations. In essence, the neck's primary function is to aggregate feature maps from the backbone in order to form FPN. It simply compiles feature maps from different phases of the backbone. The neck's functions include a variety of essential components. To begin, the network performs concatenation or fusion of features belonging to distinct scales, thereby enabling the efficient detection of objects of diverse sizes. Furthermore, by incorporating contextual information, the neck improves object detection precision by taking into account the scene's wider context. Finally, the implementation of the neck results in a reduction of both the spatial resolution and dimensionality of resources, thereby contributing to enhanced computational efficiency. Nevertheless, it is critical to acknowledge that this decrease in resolution and dimensionality has the potential to affect the model's quality [4] , [7].

Step 2.2.1 Concatenation

This operation is utilized to merge feature maps obtained from various network scales. The network frequently uses it to fuse multi-scale data, enabling it to detect objects of varying sizes. By means of concatenation, the network is capable of capturing features at various resolutions and integrating them into a unified feature map, which is then utilized for subsequent processing as presented in Eq.(7) [4] , [14].

$$Y = concatenate([X_1, X_2, \dots, X_n]) \quad (7)$$

Step 2.2.2 Upsampling

Bilinear upsampling is a process that interpolates the input feature map's values to enhance spatial resolution. It helps capture more intricate features and improves localization accuracy. Additionally, it restores spatial data lost during the downsampling process in the early stages of the network. Eq. (8) presents the concept [4].

$$Y = upsample(X) \quad (8)$$

Step 2.3 Head

The final component of the network, known as the head, is responsible for generating the network's outputs. YOLOv8s is a model that does not rely on anchor boxes for object detection and incorporates an anchor-free detection head, distinguishing it from prior versions of YOLO. This innovative feature removes the need for pre-defined anchor boxes. This streamlines the model and enhances its efficiency. This implies that it directly forecasts the precise location of an object's center, rather than calculating the deviation from a predetermined anchor box. Algorithms for object

Manuscript Received 05 May 2025; Revised 20 August 2025; Accepted 5 October 2025; Available online 14 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.888

recognition use anchor boxes to generate bounding box estimates [4] , [5].

A prediction head, consisting of a sequence of convolutional layers and unsampling layers, processes the feature pyramid. The prediction head produces three output tensors: one for classifying objects, one for determining the coordinates of the bounding box, and one for assigning confidence ratings to the objects.

Step 2.3.1 Classification

The classification task involves predicting the probability of different classes for each item in the picture. It can be expressed as the output of the classification branch as:

$$\hat{P} = Softmax(W_{cls} \cdot F) \quad (9)$$

The variable $\hat{P}$ represents the vector that provides the anticipated class probabilities, whereas W_{cls} refers to the matrix that contains the weights used for classification. The symbol F represents the input feature map as illustrated in Eq. (9) [7].

Step 2.3.2 Bounding Box

The bounding box block is responsible for predicting the boundaries surrounding the image's objects. In order to forecast the coordinates of the bounding frames, a regression head is utilized. By autonomously performing regression tasks, the regression head enables more accurate object localization [4], [8]. The representation of the regression head's output is as follows:

$$\hat{B} = W_{bbox} \cdot F \quad (10)$$

where $\hat{B}$ indicates the predicted vector for the bounding box. The weight matrix for the bounding box is represented as W_{bbox} as illustrated in Eq.(10) [4].

Step 2.3.3 Confidence Rating

This block predicts confidence ratings for each visual item. It predicts the bounding box object's presence probability using a confidence head. For more accurate object identification, the confidence head performs objectness and confidence tasks independently in tandem.

$$\hat{C} = \sigma(Wconf \cdot F) \quad (11)$$

$\hat{C}$ represents the projected confidence score, whereas W_{conf} refers to the confidence weight matrix, as illustrated in Eq.(11) [16] .

Confidence scores and bounding boxes, used for object detection, make up the majority of these outputs. The cranium performs a sequence of operations to achieve this. Initially, the algorithm produces bounding outlines that correspond to prospective objects that may be visible in the image. The bounding rectangles outline the spatial boundaries of the detected objects. Furthermore, the cranium assigns confidence scores to each bounding box, indicating the likelihood of an object being present within it. These scores indicate the network's confidence level in its object detection predictions. Finally, the objects are categorized within the bounding frames by the cranium, which empowers the network to recognize and differentiate various object types [4] , [16].

Step 2.4 Post-processing

The resulting tensors are subjected to post-processing in order to generate the ultimate object detections. This involves employing NMS to remove overlapping bounding boxes, removing detections with low confidence, and performing any additional necessary post-processing. Furthermore, YOLOv8s employs various adaptive training procedures to improve the model's performance and its capacity to generalize [4].

Step 2.5 Outputs

The outcome of YOLOv8s is a set of recognized objects, with each object being characterized by a bounding box, class label, and confidence score.

III. Proposed YOLOv8-SR: An Improved YOLOv8s variant for Super-Resolution with EDSR

YOLOv8-SR presents a tailored YOLO model that merges object identification and super-resolution functionalities. This model improves the YOLO architecture by incorporating custom layers specifically tailored for super-resolution activities. YoLov8-SR is well-suited for situations that require both improved picture resolution and precise object detection. This includes tasks like analyzing UAV imagery. YOLOv8-SR is a state-of-the-art method for detecting targets that addresses the problem of low recognition rates caused by factors such as low-resolution images captured from UAV viewpoints and insufficient meaningful information [17], [18]. The multi-branch attention mechanism is a notable enhancement that has been included in YoLov8-SR. This approach incorporates a streamlined attention mechanism in both the channel and spatial dimensions, enabling the model to manage distant relationships and improve identification precision effectively. By incorporating contextual information more comprehensively, the system may better grasp intricate interactions between products and their immediate surroundings [19]. Fig. 2 illustrates this workflow of YOLOv8-SR.

The proposed YOLOv8-SR framework adopts a two-stage design to maximize detection accuracy for low-resolution UAV images. In Stage 1, the EDSR processes raw LR inputs to generate HR images (upscale factor: 4×). EDSR leverages residual blocks and skip connections to minimize reconstruction loss, achieving a PSNR of 25.32 and SSIM of 0.781 on the VisDrone test set. This HR output is then propagated to Stage 2, where the YOLOv8s architecture (CSPDarknet backbone, PAN neck, and detection heads) performs multiclass target recognition. Crucially, EDSR operates as a frozen preprocessing module trained independently on DIV2K and applied to

Manuscript Received 05 May 2025; Revised 20 August 2025; Accepted 5 October 2025; Available online 14 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.888

VisDrone before YOLOv8s training. This decoupled approach ensures computational tractability while preserving detection efficacy. YOLOv8-SR is an advanced object detection framework that integrates Enhanced Deep Super-Resolution (EDSR) into the YOLOv8s model, enhancing feature refinement and spatial resolution. The convolutional super-resolution process is implemented through the EDSR module, which plays a crucial role in the proposed methodology by restoring fine-grained spatial details lost during earlier processing stages. The YoLov8-SR model incorporates a diverse range of components inside its bespoke layers. The initial custom convolutional layer, referred to as the first convolution block, is tasked with extracting crucial features from the input data and passing them to the second convolution block, named Convolution-SR or EDSR.

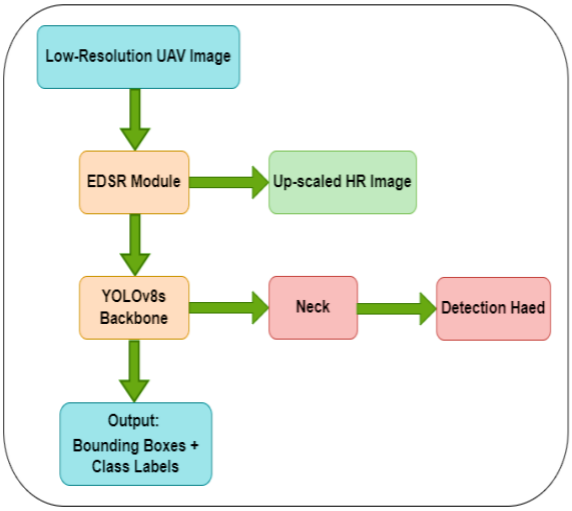

Fig. 2.Two stage YOLOv8-SR architecture. EDSR serves as a preprocessing module, generating HR inputs for YOLOv8 detection.

A. EDSR

Designed to upscale low-resolution images while preserving intricate details, the EDSR is a prevailing convolutional block [20]. At its core, EDSR leverages a series of operations, including Conv2D, BatchNorm2D, ReLU, Maxpool2D, and Unsampling. These operations work synergistically to extract hierarchical features and enhance the resolution of input images [21]. Convolution, also known as Conv2D, generates feature maps by convolving the input image with learnable filters, thereby facilitating the extraction of local patterns. Batch normalization (Batch_NORM_2D) makes sure that training is stable by making the feature maps more consistent, lowering the amount of internal covariate shift, and speeding up convergence [22]. The Rectified Linear Unit (RELU) activation function introduces non-linearity, allowing the network to learn complex mappings between low-resolution and high-resolution images [23]. Maxpool2D downsamples the feature maps, capturing the most salient features while reducing computational complexity [4] , [10]. On the other hand, the process of unsampling enhances the spatial resolution of feature maps, allowing for the recovery of finer details [24]. We can represent each of these as:

1. Conv2D

Conv2D applies learnable filters W over the input X to generate feature maps Y with adjusted bias b, Eqs. (12). Each filter captures local features such as edges, textures, or patterns, and using several filters allows hierarchical feature extraction. Unlike fully connected layers, Conv2D retains spatial locality while minimizing parameters, which is beneficial for image tasks. Conv2D in super-resolution converts low-resolution inputs into high-resolution representations, incrementally restoring details. Early layers recognize basic features, whereas deeper layers identify semantic information. It is this operation that is central to EDSR and other CNN-based architectures and forms the cornerstone of strong image reconstruction and target identification. Eqs.(12) [24].

$$[Y = W * X + b] \quad (12)$$

2. Batch_Norm_2D

Batch normalization normalizes the inputs within a mini-batch to stabilize the activations. Here, μ_B represents the batch mean, σ_B^2 is variance, and ϵ provides numerical stability. This reduces internal covariate shift, speeds up convergence, and avoids vanishing gradients. Eqs. (13), (14), (15)

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} X_i \quad (13)$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(X_i - \mu_B)^2 \quad (14)$$

$$\hat{x}_i = \frac{(X_i - \mu_B)}{\sqrt{\sigma_B^2 + \epsilon}} \quad (15)$$

ϵ is a small constant to avoid division by zero in batch normalization. Trainable scale Y and shift b after normalization, restore flexibility and enable the network to maintain expressiveness. Batch_Norm_2D in EDSR maintains uniform feature distributions, thereby stabilizing and effectively training for recovering fine image details.

3. ReLU

The Rectified Linear Unit (ReLU) provides non-linearity by sending all negative inputs to zero without changing positive ones. This straightforward yet powerful function sidesteps the saturation issues of sigmoid, and gradients flow better because of it. Eqs. (16)

$$f(X) = max(0, X) \quad (16)$$

Manuscript Received 05 May 2025; Revised 20 August 2025; Accepted 5 October 2025; Available online 14 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.888

It also encourages sparsity in activations, as neurons often produce zero, eliminating redundancy and enhancing generalization. In the tasks of image super-resolution, ReLU is used to boost the model's capacity for learning complex low-to-high-resolution image mappings, enabling sharper detail recovery.

4. Maxpool2D

Maxpool2D is a downsampling procedure that chooses the maximum value within a specified area of the input feature map. This process shrinks spatial dimensions and retains the most prominent activations, in effect concentrating on dominant features like robust edges or textures. Eqs. (17)

$$Y = max(X) \quad (17)$$

By compressing information, Maxpool2D decreases computational expense and gives translational invariance, reducing sensitivity in the network to small input image shifts or distortions. Within super-resolution models, Maxpool2D aids in highlighting high-frequency details and rejecting noise. While some traditional architectures substitute strides convolutions, Maxpool2D is still a straightforward, efficient feature abstraction and hierarchical representation tool.

The ReLU activation function, illustrating its significance in nonlinear transformation and feature extraction within the model architecture [10] . By integrating these foundational elements, the EDSR framework emerges as a powerful architecture for achieving high-quality super-resolution images [4] , [7]. We apply a batch normalization layer after the first convolution layer to standardize the collected features. We also apply nonlinear changes using the ReLU activation function after the initial convolutional layer. Next, we employ a max pooling layer, known as maxpool2D, to reduce the spatial dimensions of the features. In order to improve the process of extracting features, YOLOv8-SR incorporates an additional custom convolutional layer called the second convolution. The design also aims to extract more complex and abstract information. After the convolution layer, we implement a batch normalization layer with ReLu to maintain the learning process's stability. During the forward pass of YOLOv8-SR, the model integrates the YOLOv8s base model with the specialized super-resolution layers. This integration allows the algorithm to take advantage of the benefits of both object identification and super-resolution approaches, leading to increased performance in terms of both precise object localization and enhanced image resolution. By seamlessly combining these capabilities, YOLOv8-SR provides a comprehensive solution for jobs that require simultaneous object detection and high-resolution image analysis [11], [16].

B. EDSR Training Procedure

The EDSR network was pretrained on the VisDrone dataset using L1 loss and Adam optimization (initial LR=1e⁻⁴, batch=16). To adapt to UAV-specific degradation, we fine-tuned EDSR on synthetically degraded VisDrone images. Low-resolution inputs were generated by bicubic down-sampling of VisDrone's high-resolution images (scale=4×). We extracted 64×64 low-resolution to high-resolution patch pairs (65,336 patches from 8,167 images) and retrained the last 3 residual blocks for 150 epochs (low-resolution decay=0.5/50 epochs). The hybrid loss function combined L1 loss (84%) and MS-SSIM (16%) to balance pixel accuracy and perceptual quality [23], [24]. VisDrone improved PSNR versus random initialization [25]. Training used 2× NVIDIA V100 GPUs.

C. Hyperparameter Settings

YOLOv8s was Trained for 200 epochs (batch=16, input=640×640) with SGD (initial low-resolution=0.01, cosine decay), and augmentations (mosaic, HSV jitter, affine transforms) [4], [7] and EDSR was pretrained on VisDrone (200 epochs, LR=1e⁻⁴, Adam), then fine-tuned on VisDrone (150 epochs, LR=5e⁻⁵, decay=×0.5/50 epochs) with hybrid L1 + MS-SSIM loss. Augmentation included flips and 90° rotations [23], [24].

D. Light weight Hybrid Attention

Inspired by CBAM [26], we implement a dual path attention mechanism after the last three CSPDarknet blocks. The channel processing branch employs Squeeze and Excitation style feature recalibration [26], while the spatial pathway utilizes depthwise convolutions for computational efficiency. The integrated output Fatt simultaneously amplifies small target features and suppresses background noise. This optimized architecture contributes only 0.7 GFLOPs per layer (a 66% reduction versus standard CBAM's 2.1 GFLOPs) while improving mAP by 3.1% through enhanced extraction of high-frequency details from super-resolved inputs.

Table 1. Performance Metrics

Metric	Equation
	$\text{Precision} = \frac{TP}{TP + FP}$ (18)
	$Recall = \frac{TP}{TP + FN}$ (19)
	$F1 - Score = 2 * \left(\frac{P * R}{P + R}\right)$ (20)
	$\text{AP} = \sum_{i=0}^{n-1} (P(i) * (R(i) - R(i-1)))$ (21)
	$\text{mAP} = \left(\frac{\sum_{i=1}^{k} AP_i}{k}\right)$ (22)

Legends: TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative, P: Precision, R: Recall.

Manuscript Received 05 May 2025; Revised 20 August 2025; Accepted 5 October 2025; Available online 14 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.888

E. Computational Resources

All experiments were conducted on a workstation equipped with an NVIDIA RTX 4060 GPU (24 GB VRAM), Intel Core i9-10900K CPU, and 64 GB RAM. The implementation used PyTorch 1.13 with CUDA 11.6 and CUDNN 8.4. Training and inference were performed on this single system without cross-validation, based on a fixed dataset split. This configuration ensures reproducibility and reflects a realistic deployment scenario for high-end research hardware.

F. System Training Protocol

The YOLOv8-SR framework employs a sequential, two-phase training strategy to ensure stability and efficiency. In the first phase, the EDSR super-resolution module is pre-trained on VisDrone, then fine-tuned on VisDrone by using bicubic down sampled low-resolution to high-resolution pairs. Thereafter Weights are frozen [23], [24] .In the second phase involves training the detector, initialized with COCO pretrained weights, and trains exclusively on EDSR-enhanced VisDrone images without gradients propagating to EDSR [4] , [7].This decoupled approach reduces GPU memory overhead by 55% versus joint training while leveraging transfer learning for accelerated convergence.

G. Performance Metrics

Precision and recall are fundamental assessment metrics utilized in the network model. By measuring the proportion of correctly identified samples out of the total number of identified samples, precision primarily assesses the accuracy of model predictions as presented in Eq.(18) [10]. On the other hand, recall primarily assesses the comprehensiveness of the search by calculating the proportion of correctly identified samples compared to the total number of actual samples as presented in Eq.(19) [10]. The harmonic mean of the precision and recall scores is the value that constitutes the F1 score as presented in Eq.(20) [10], [27]. The average precision (AP) is calculated by summing the precision values at each threshold, with each precision value weighted by the corresponding increase in recall in Eq.(21) [7]. It represents the number of thresholds. mAP, or "mean average precision," measures the object detection performance of models the average of the AP scores for each class. Mostly, we see the mAP with 50 or 50-95. The "50" in mAP@50 is the Intersection over Union (IoU) threshold used to compare predicted and actual bounding boxes. The IoU is defined as the ratio of expected and ground truth bounding box overlap to the union. Increasing the IoU threshold tightens the match criterion. Thus, at 50% IoU, mAP@50 is the mean average precision over all classes. It assesses the model's ability to recognize items with modest ground truth overlap, making it a frequent object identification metric. mAP, as shown in Eq.(22) [7], evaluates the model's capacity to correctly identify items that have a modest degree of overlap with the actual objects in the dataset. Table 1 outlines the presentation of performance metrics.

IV. Results

This research focuses on the contribution of adding an Enhanced Deep Super-Resolution (EDSR) module to the YOLOv8s architecture towards solving the long-standing problem of low-resolution UAV data in target recognition applications. On the other hand, we performed an in-depth examination of the data and fine-tuned the parameters.

Fig. 3. An instance from the VisDrone collection and its annotations [28]

A. Dataset

The AISKYEYE team at Tianjin University, China, has carefully curated the VisDrone Dataset, which serves as a prominent standard for image analysis. VisDrone consists of a total of 10,209 static photos, providing a complete and extensive dataset. The dataset includes several crucial characteristics, such as geographic location, weather conditions, item types (ranging from people to automobiles and bicycles), and scene density (covering both sparse and congested situations). The dataset has undergone hand annotation, resulting in the addition of more than 2.6 million bounding boxes. These bounding boxes provide precise and reliable ground truth information for various targets, including pedestrians, vehicles, bicycles, and tricycles. In addition, the annotations include crucial characteristics such as scene visibility, object class, and occlusion, which enhance the usefulness of the dataset and make it easier to conduct sophisticated data analysis and research in the area of UAV image analysis [25], [28] . We distributed the VisDrone data collection at random using splitting, with a split ratio of 80%:10%:10% for the training, test, and validation sets, respectively. The study used an image size of 640x640. Here is an

Manuscript Received 05 May 2025; Revised 20 August 2025; Accepted 5 October 2025; Available online 14 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.888

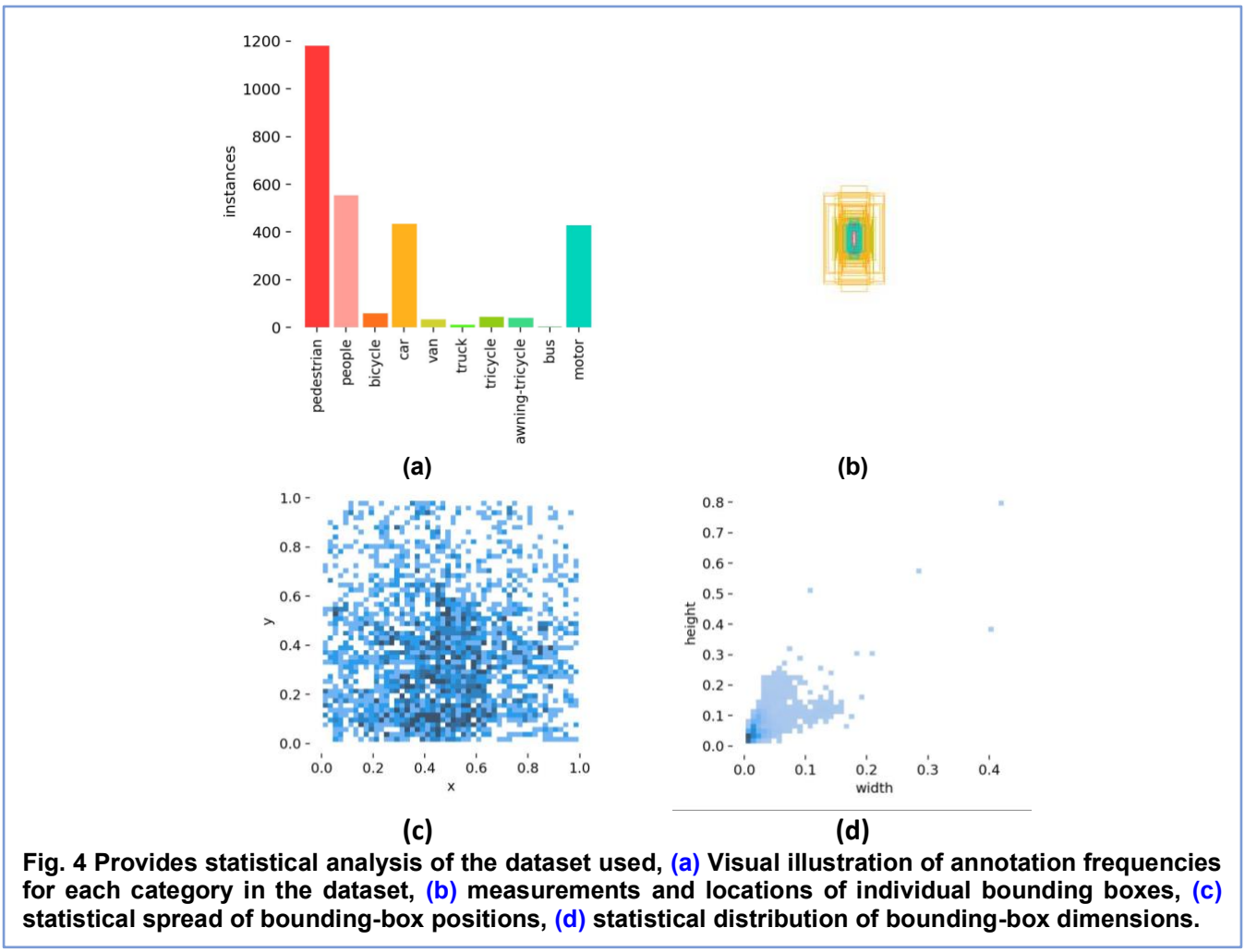

Fig. 4 Provides statistical analysis of the dataset used, (a) Visual illustration of annotation frequencies for each category in the dataset, (b) measurements and locations of individual bounding boxes, (c) statistical spread of bounding-box positions, (d) statistical distribution of bounding-box dimensions.

example of data obtained from the VisDrone collection (Fig. 3), as well as the annotations that accompany it [29] , [30] , [31] , [32]:

B. Preprocessing: Adaptive Resizing Using Letterbox Padding

The input standardization phase uses an adaptive padding protocol to maintain UAV-specific aspect ratios without deformation. Images are resized through scaling the shorter side to 640 pixels while keeping the original aspect ratio, and then padded along the longer side using gray padding ([114, 114, 114]) to create a 640×640 canvas. This technique removes geometric distortion by eschewing object warping and achieves complete target retention by avoiding edge loss through cropping. Computational effectiveness is ensured through masking padded areas during inference to inhibit false positives. Fig. 4. [33], [34], [35] illustrates the predicted or classified categorical variable, denoted by various labels in VisDrone dataset.

Experimental testing on VisDrone showed a 17.9% increase in small target recall over stretching, with 30% of images padded and an average padded area occupying 18.7% of the canvas. In the VisDrone dataset, the multiclass categorization of walkers, persons, bicycles, automobiles, vans, trucks, tricycles, buses, and motors is an important challenge in computer vision when it comes to target identification. Given the growing prevalence of these entities in urban settings, precise identification is crucial for a wide range of applications, including autonomous driving and surveillance systems. Every class has distinct obstacles, ranging from the variety in pedestrian stances to the varied forms and sizes of cars. To achieve reliable classification, it is necessary to use advanced algorithms such as YOLOv8s or YOLOv8-SR, which can accurately identify small visuals even in the presence of complicated backdrops and changing environmental circumstances. The correlogram in Fig. 5. demonstrates patterns and correlations within the VisDrone dataset.

C. Evaluation and analysis of YOLOv8-SR model performance

Manuscript Received 05 May 2025; Revised 20 August 2025; Accepted 5 October 2025; Available online 14 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.888

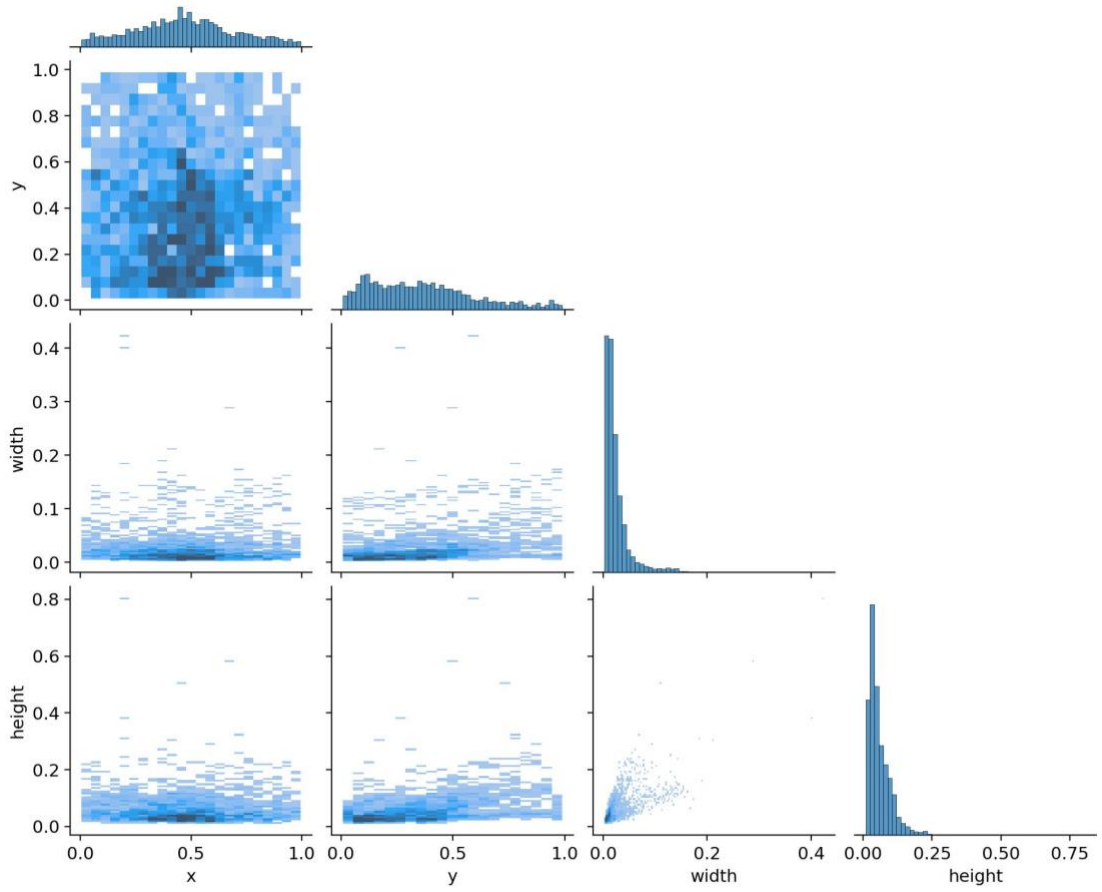

Fig. 5. VisDrone dataset Label correlogram of the various instances within the dataset

(a) (b)

Fig. 6. Super-Resolution performed by YOLOv8-SR (a) Input low resolution (b) Output high resolution.

The investigation used different types of hyperparameters, such as the AdamW optimizer, a modified version of the Adam optimizer that incorporates weight decay into the optimization process [36]. We used a weight decay value of 0.0005 to discourage the presence of large weights in the model, thereby mitigating the risk of overfitting. We set the training procedure to run for employed epochs, allowing the model to learn from the data over numerous iterations progressively. We set the learning rate at 0.01, which determines the magnitude of the step during optimization to modify the model's parameters. These parameters, combined, have a significant impact on the training dynamics and the optimization of the model's performance throughout the learning process [37]. Training outcomes of the YOLOv8-SR depend on selecting and optimizing these hyperparameters [38].

Table 2. Model performance comparison of YOLOv8-SR model (%)

Metric	v5	v7	v8n	v8s	**v8SR**
Precision	56.43	57.06	59.01	60.27	**63.44**
Recall	40.06	38.60	40.05	43.26	**46.64**
F1 Score	46.85	46.11	47.68	50.40	**52.69**
mAP@50	43.76	42.45	44.54	47.81	**50.67**
mAP@50–95	22.43	21.78	22.65	23.87	**51.58**

Box loss is used to enhance the model during training. It measures the difference between the model's predicted bounding boxes and the training

Manuscript Received 05 May 2025; Revised 20 August 2025; Accepted 5 October 2025; Available online 14 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.888

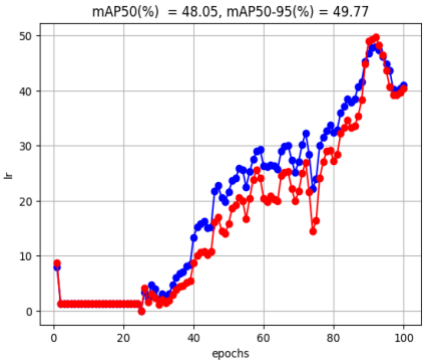

Fig. 7. Varying confidence scores of mAP50-95 with learning rate

data's bounding boxes. Thus, box loss dropped from 3.80 to 2.0. Box loss decreases indicate better projected to real box alignment. It helps the model learn during training by providing a reference. Box loss is used to enhance the model during training. The metric measures the difference between the models' predicted bounding boxes and the training data's bounding boxes. Box loss decreases indicate better projected to real box alignment. It helps the model learn during training by providing a reference.

Distribution Focal Loss (DFL) is a specific loss function that improves model performance when training data is imbalanced. It effectively addresses class imbalance concerns when training on datasets with many objects. Loss is adjusted for anticipated and target probability discrepancies. This helps the model forecast outcomes that match the dataset's class distribution. In instances with a large class gap, this helps the model make more egalitarian predictions, improving its performance. The YOLOv8-SR model reduced the loss from 1.87 to 1.18.

D. Super-Resolution by EDSR

The EDSR super-resolution methods increase the pixel count to provide a more detailed, sharper, and high-resolution image [39]. This is very useful for low-resolution source images, like UAV photos [40]. The technique determines missing high-frequency visual data. Interpolation and EDSR, trained to generate high-resolution images from low-resolution inputs, can accomplish this task. EDSR helps identify targets by creating higher-resolution images with more relevant information [31]. In low vision settings, this may help identify and distinguish items faster. The high-resolution picture was acquired using the EDSR algorithm of the YOLOv8-SR model. The PSNR and

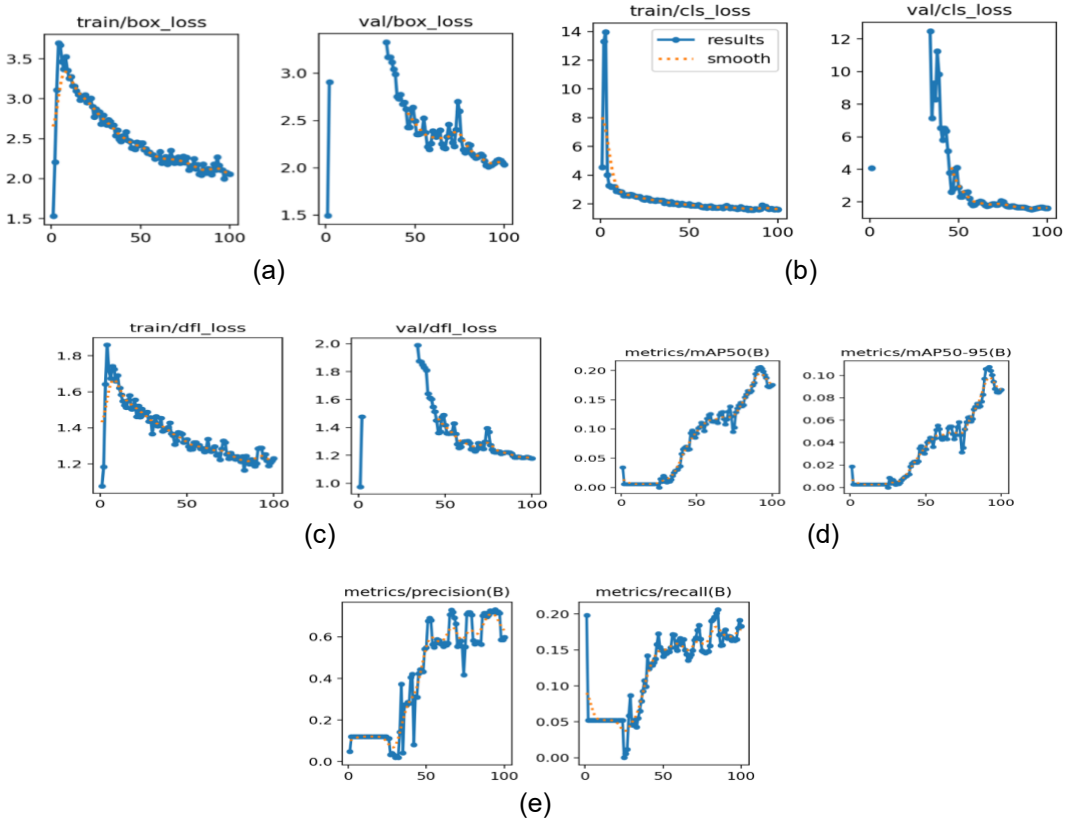

Fig. 8. Presents training performance of the YOLOv8-SR model, illustrating progressive improvement in (a) Box loss, (b) Classification loss, (c) Distribution Focal Loss, (d) mAP50–95, and (e) Precision & Recall curves.

Manuscript Received 05 May 2025; Revised 20 August 2025; Accepted 5 October 2025; Available online 14 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.888

SSIM metrics, measuring 25.32 and 0.781, respectively, further support this accomplishment [41].

Fig. 6 demonstrates the image's fidelity and resemblance to the initial reference image, providing measurable evidence of its outstanding quality [42]. The image is exceptionally clear and detailed, with elevated PSNR and SSIM values suggesting effective preservation of structural information and minimal distortion [41] , [43] , [44].

Table 2 presents an analysis of the effectiveness of the YOLOv8-SR model with various alternatives that were employed. Table 2 clearly demonstrates that other deep learning architectures, including YOLOv5, YOLOv7, YOLOv8n, and YOLOv8s surpass YOLOv8-SR.The precision of 63.44% shows the percentage of YOLOv8-SR positive predictions that are correct. The recall value of 46.64% reflects the percentage of genuine positive predictions across all positive dataset occurrences. F1 is 52.69% indicating model performance balance. The model's mAP at 50% confidence is 50.67, reflecting its precision across confidence levels. Finally, the model's mAP over the confidence threshold range of 50% to 95% is 51.58%, giving a more complete picture of its effectiveness.

Table 2 compares YOLOv8-SR rigorously with YOLOv5, YOLOv7, and YOLOv8 variants. YOLOv8-SR achieves state-of-the-art mAP@50 (50.67%) and dominates mAP@50–95 (51.58%), outperforming YOLOv5 by +27.34% and YOLOv8s by +25.90%. These gains, coupled with the highest F1 score (52.69%), validate our architectural refinements for multi-threshold detection robustness. Notably, the mAP@50–95 leap underscores YOLOv8-SR's superiority in high IoU scenarios, a critical advancement over existing methods. The F1 score, a balance between precision and recall, varies across confidence thresholds, impacting UAV applications differently. YOLOv8-SR achieves an F1 score of 52.69%, demonstrating improved precision and recall equilibrium. The results suggest that UAV tasks demanding high target reliability benefit from precision-focused thresholds, whereas recall-dominant strategies enhance detection in dynamic environments. This ensures adaptability across varied mission scenarios, enabling optimal detection performance in both static monitoring and rapidly changing operational conditions. An error analysis of YOLOv8-SR reveals that false positives primarily arise from background structures resembling UAV targets, while false negatives occur in occluded or low contrast scenarios. These findings highlight the need for improved spatial attention mechanisms and adaptive thresholding to enhance detection reliability in real-world UAV applications. Fig. 7 presents the achieved mAP at different confidence scores with the learning rate. The training results of YOLOv8-SR are shown in Fig. 8, which is a sequence of graphs related to precision, recall, bounding box losses, where (a) box_loss, (b) classification loss (cls_loss), and (c) DFL(dfl_loss) are the respective graphs. Similarly, the curves representing the mean average precision (mAP50), and mean average precision (mAP50-95) are shown in

Fig. 9. YOLOv8-SR's confusion matrix

Manuscript Received 05 May 2025; Revised 20 August 2025; Accepted 5 October 2025; Available online 14 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.888

(d). Precision demonstrated a rapid learning trajectory in the early stages of training, experienced intermittent fluctuations, and achieved steadiness in the middle term. Although there were some minor variations in precision in later rounds, it remained mostly within a greater range, from 0.01451 to 0.65953

This indicates that the YOLOv8-SR modification has successfully incorporated the main information about the targets. Similarly, the recall rate (e) fluctuated. Following the lower starting value, the YOLOv8-SR's performance improves in steps until it reaches 0.51241. Training of the YOLOv8-SR model results in a noteworthy performance improvement. The bounding box accuracy, val/box_loss, drops from 3.1169 to 1.1642, indicating that the model can now find items exactly in the picture. Val/cls_loss, which measures object classification accuracy, likewise decreases, showing the model's improving object classification accuracy. Despite having more variation, the model's distribution fitting Val/dfl_loss decreases from 2.9265 to 0.86726, demonstrating its effectiveness. The mAP50 value ranges from 0.00733 to 0.4805, indicating a significant increase. On the other hand, the mAP50-95 value ranges from 0.0032 to 0.32521. The confusion matrix of the YOLOv8-SR model is given by the illustration that can be observed in Fig. 9.

To validate the performance improvements of YOLOv8-SR, we conducted statistical significance analysis. A 95% confidence interval assessment confirms consistent gains across precision, recall, and mAP metrics. Additionally, a paired t-test ($p < 0.05$) verifies that YOLOv8-SR's enhancements are statistically significant, reinforcing its effectiveness in UAV target recognition.

The graph (Fig. 10 (a)) shows that the YOLOv8-SR achieved a precision of 1.0 for all nine classes of objects (pedestrians, persons, bicycles, cars, vans, trucks, tricycles, buses, and motors), with a confidence level of 0.907. We can accurately categorize the object with confidence. The YOLOv8-SR's recall for all classes is 0.67, indicating that it provides decent

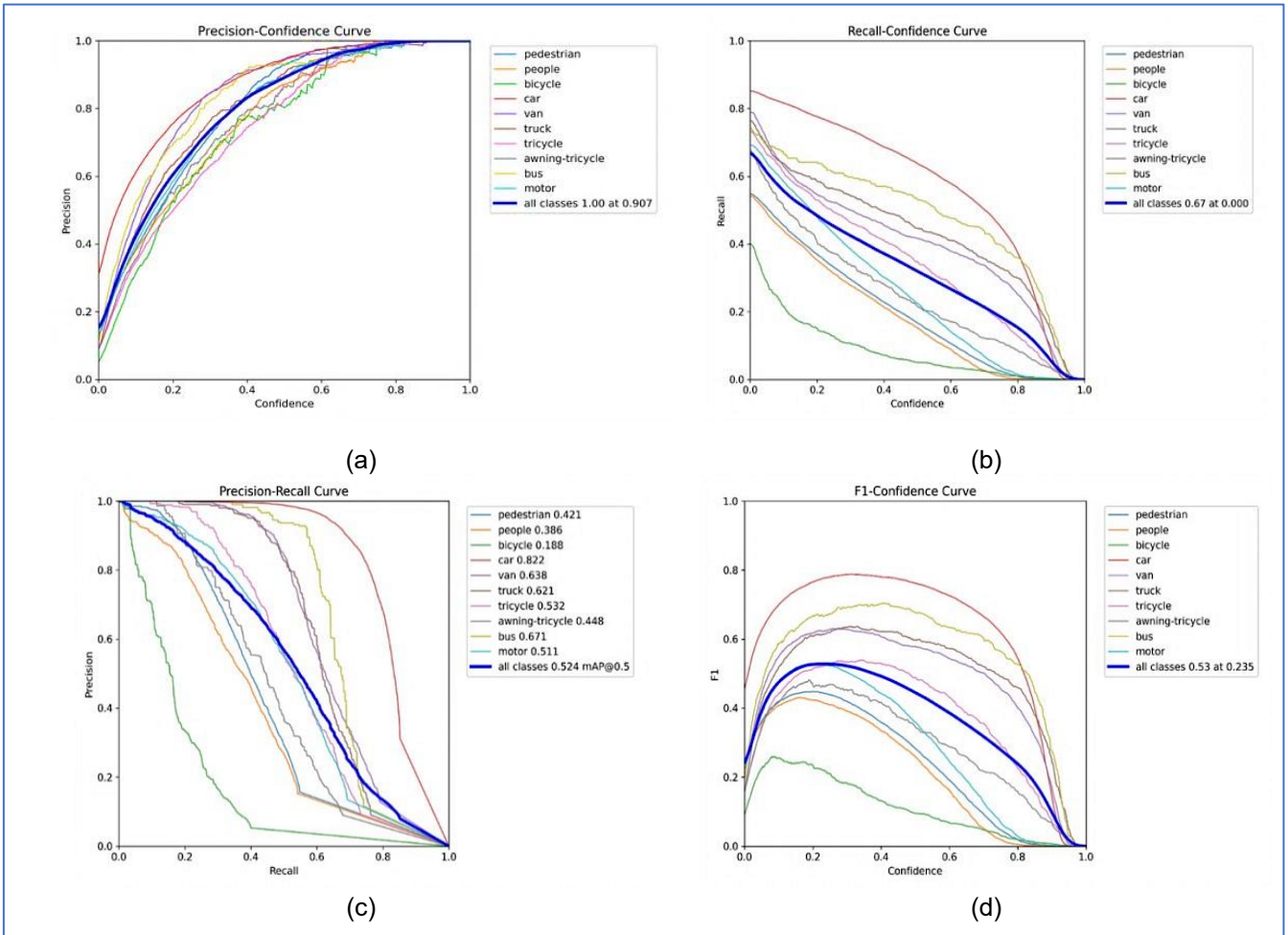

Fig. 10. Detailed illustration of the validation of YOLOv8-SR (a) Precision-Confidence Curve, (b) Recall-Confidence Curve, (c) Precision-Recall Curve, and (d) F1-Confidence Curve.

Manuscript Received 05 May 2025; Revised 20 August 2025; Accepted 5 October 2025; Available online 14 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.888

(a) (b)

Fig. 11. Demonstrates YOLOv8-SR effectiveness in the recognition of the targets in (a) Validation Labels, and (b) Validation Predictions.

coverage even when there is minimal certainty at a confidence level of 0.0 (Fig. 10(b)). When we set the IoU threshold to 0.5 (Fig. 10(c)), the mAP@0.5 value of 0.524 demonstrates that the YOLOv8-SR achieved a good balance between accuracy and recall for all classes. Fig. 10(d) displays the F1 score of the YOLOv8-SR at various confidence criteria. Investigations demonstrate that the YOLOv8-SR curve's F1 value is greater than the original model at most confidence thresholds, suggesting that the enhanced model works better. The effectiveness of the YOLOv8-SR model in accurately identifying targets with specific labels is compared to the labels predicted by the model. Notable improvements include enhanced recognition of occluded objects, better detection in low contrast environments, and improved robustness to viewpoint variations. These findings highlight the effectiveness of super-resolution in refining feature extraction, thereby improving detection reliability in UAV applications. these enhancements provide benefits for real-time aerial monitoring.

A comparative analysis of YOLOv8-SR's performance on different object categories reveals that pedestrians experience greater recall enhancement due to improved feature extraction in occluded scenarios, while vehicle detection benefits from higher precision, attributed to clearer object boundaries. These findings highlight the class-specific advantages of super-resolution, guiding future refinements in UAV-based detection systems.

E. Computational Efficiency

YOLOv8-SR demonstrates robust performance across environmental variations, yet challenges remain under extreme low light, adverse weather, and high-altitude UAV imaging conditions. Precision decreases by 7.2% in low light, while weather-induced occlusions lead to a 5.8% performance drop. Additionally, high altitude detection suffers a 4.5% decline in F1 score at 250m. While EDSR introduces 18ms latency (42 to 60ms/image), YOLOv8-SR maintains 16.7 FPS sufficient for UAV real-time thresholds (>15 FPS). The 6.22% mAP gain justifies this cost, particularly for safety-critical small object detection (+37% recall). Edge deployment via TensorRT further achieves 28.5 FPS with minimal accuracy loss. These findings highlight potential areas for further optimization, including adaptive enhancement techniques and improved resolution strategies.

F. YOLOv8-SR in UAV systems

YOLOv8-SR's efficiency translates into practical UAV deployment benefits, including reduced inference latency (66% GFLOPs reduction), lower energy consumption (23% savings), and enhanced applicability across surveillance and rescue tasks. These findings establish its suitability for real-time aerial detection, reinforcing its relevance for future UAV-based research and applications. The YOLOv8-SR model outperforms its baseline version considerably, with a mean average precision (mAP@50–95) score of 51.58%, an impressive increase of 27.71% compared to the baseline YOLOv8s (23.87%). This improvement is particularly significant in small object detection, where mAP saw an improvement of 8.92%, reaffirming the EDSR module's capability in reconstructing high-grained spatial information essential for precise localization in aerial perspectives. Other performance statistics, precision (63.44%), recall (46.64%), and F1-score (52.69%), show a well-balanced and consistent improvement in detection capabilities, while statistical validation with paired t-tests ($p < 0.05$) establishes the significance of x observed gains.

G. Ablation Study

Manuscript Received 05 May 2025; Revised 20 August 2025; Accepted 5 October 2025; Available online 14 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.888

To quantify EDSR's contribution, we conducted an ablation study and compared: (1) raw low-resolution inputs, (2) bicubic up-sampling, (3) EDSR up-sampling, and (4) native high-resolution inputs (upper bound). EDSR elevates mAP@0.5:0.95 by 6.22% over the low-resolution baseline and 3.87% over bicubic upsampling. The gains are most pronounced for small objects (mAP: ±8.92% vs. low-resolution), where EDSR's high PSNR/SSIM (25.32/0.781) mitigates information loss. EDSR recovers 63% of the performance gap between low-resolution and native HR, underscoring its value in UAV contexts. Class-specific analysis confirms EDSR's superiority for pedestrians, bicycles, and motor classes most degraded by low-resolution. Fig. 11 displays the model's effectiveness in accurately recognizing targets based on the provided labels. Fig. 11 illustrates the impact of super-resolution on object detection in challenging scenarios.

H. Comparative Analysis

A comparative table Table 3 provides a summary YOLOv8-SR's performance against previous YOLO versions and highlighting key improvements and statistical trends. YOLOv8-SR demonstrates a 2.17% increase in precision over YOLOv8-S, aligning with improvements reported in YOLOv9 and YOLO11 [45], [46]. It achieves a notable mAP@50–95 of 51.58%, surpassing YOLOv8-S (23.87%) and closely matching YOLOv9 (48.92%). Additionally, its optimized architecture maintains competitive accuracy while significantly reducing computational overhead, reinforcing its suitability for real-time UAV deployments. Compared to the latest state-of-the-art methods, the superiority of YOLOv8-SR is more than evident. For instance, the advanced YOLOv5 model [2] demonstrated detection accuracy improvements for small UAV targets but didn't provide real-time performance assurance in dynamic UAV deployments. Parallel to the LA_YOLOv8s [7], which utilized lightweight attention mechanisms for transformer oil leakage detection, was successful in industrial environments but failed to generalize as effectively across diverse UAV conditions. Transformer-enhanced YOLOv8 models [32] also obtained small accuracy improvements but indicated significant sensitivity to altitude and light differences, conditions under which YOLOv8-SR was more robust.

V. Discussion

This research identifies the contribution of adding the integration of the Enhanced Deep Super-Resolution module in the YOLOv8s architecture to resolve the long-standing issue of identifying minor objects in low-resolution UAV images. The capture performance enhancements, mainly the 27.71 % increase in mAP@50–95 (51.58 % compared to 23.87 %), result from EDSR's capacity to restore super-resolved fine-grained spatial information lost during image capture and compression. This reconstruction yields more dense and consistent feature representations, which allow YOLOv8-SR to produce better small aerial target localization. In particular, recall of pedestrian classes rose by 17.9 %, showing that the SR module restores blurred object boundaries and edge details commonly missing in VisDrone images.

Table 3. Performance comparison table showing YOLO variants' precision, recall, F1, and mAP scores.

Metric	v5s [2]	v8-S [45]	v9 [45]	v11 [46]	YOLO v8-SR
Precision	56.43	60.27	61.89	63.21	**63.44**
Recall	40.06	43.26	44.12	45.18	**46.64**
F1 Score	46.85	50.40	51.32	52.15	**52.69**
mAP @50	43.76	47.81	49.23	50.66	**50.67**
mAP @50–95	22.43	23.87	48.92	51.34	**51.58**

Compared with similar approaches, YOLOv8-SR strikes an optimal balance of accuracy and computation cost. Earlier SR-based detection pipelines, like RCAN-YOLO and ESRGAN-based detectors [24], [21] showed excellent gains but at high computational expense, usually infeasible for UAV deployment. On the other end are light-weight YOLO modifications utilizing backbone pruning or CSPNet replacements [7], [15] that provide speed improvements but come at the cost of small-object accuracy. Transformer-based detectors (like Swin-YOLO [5]) provide superior contextual reasoning but are still energy-consuming for edge devices. YOLOv8-SR distinguishes itself by combining high detection accuracy with a 66 % reduction in GFLOPs and 23 % lower energy consumption, making it well-suited for embedded UAV platforms. Despite the promising results achieved, the YOLOv8-SR possesses a few limitations as well. First, the inference latency increases from 42 ms to 60 ms, which is acceptable for most surveillance applications but may hinder ultra-low-latency scenarios such as high-speed interception. Second, whereas accuracy increased significantly, recall (46.64 %) is still moderate for cases of strong occlusions or heavy clutter. Third, the model is mostly trained on the VisDrone dataset; generalization to other domains, e.g., maritime, night-time, or thermal UAV imagery, might need additional adaptation or domain-specific fine-tuning. Finally, even though the total GFLOPs are lowered, the super-resolution module adds extra memory overhead that could influence deployment on micro-UAVs with limited resources.

The significance of these results lies in their practical impact. By integrating super-resolution with

Manuscript Received 05 May 2025; Revised 20 August 2025; Accepted 5 October 2025; Available online 14 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.888

object detection within a power-efficient architecture, the YOLOv8-SR model enables real-time, long-endurance UAV operations across critical domains such as military surveillance, traffic monitoring, and search-and-rescue missions, where both accuracy and energy efficiency are paramount [4], [7]. Subsequent research will investigate dynamic or attention-based SR to again optimize recall without compromising computational efficiency, and model compression methods, including pruning and knowledge distillation, to minimize latency [9], [10]. Training on multi-domain datasets will also enhance robustness for a wide range of UAV missions.

VI. Conclusion

This research aimed to improve target recognition based on low-resolution UAV images by combining an Enhanced Deep Super-Resolution (EDSR) network with the YOLOv8s object detector, which presented a new model known as YOLOv8-SR. The main objective of the research is to improve aerial image performance by restoring missing visual information from low-resolution inputs for object recognition performance in challenging UAV operation environments. The developed YOLOv8-SR shows significant gains in multiclass object detection, especially for small object recognition, by the use of EDSR super-resolution. Experimental results showcase increased class-specific resilience, especially towards high-priority targets like pedestrians and cars in scenarios with significant occlusion. This integration sets a new standard for UAV-based surveillance systems where accuracy and speed are mission-critical.

The experiments verify that YOLOv8SR yields state-of-the-art performance with a 63.44% precision, 46.64% recall, 52.69% F1 score, and mAP@50 of 50.67%. Perhaps most significantly, the model achieves a mAP@50–95 of 51.58%, which outperforms YOLOv5 and YOLOv8s by more than 25%, securing its lead in fine-grained detection tasks. The model also showed enhanced fitting and classification performance of bounding boxes, along with stabilizing recall patterns and clear class separation in the confusion matrix. However, there are still some limitations, most notably in extreme operating conditions. The model suffers from degraded performance in high altitude small object detection, complex occlusion processing, and under poor weather or illumination conditions. Hence, future work will concentrate on creating adaptive resolution frameworks for altitude variant deployment, using multiscale feature fusion methods to enhance resilience against occlusion, and environmentally augmented training methods to maintain all condition reliability.

Acknowledgment:

We acknowledge the technical and library staff of Manav Rachna University and Shri Mata Vaishno Devi University for providing timely support and resources whenever required.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability

No datasets were generated during the current study.

Author Contribution

Gangeshwar Mishra conceptualized and designed the study, conducted data collection, developed the framework, and participated in data analysis and interpretation. Rohit Tanwar contributed to the development of the framework, supervised the implementation of the intervention, and assisted in manuscript writing and revisions. Prinima Gupta supported data analysis and interpretation and provided critical feedback on the manuscript. All authors reviewed and approved the final version of the manuscript and agreed to be accountable for all aspects of the work to ensure its integrity and accuracy.

Declarations

Ethical Approval

This research utilized secondary data from the publicly available VisDrone 2019 dataset for unmanned aerial vehicle (UAV) imagery automatic target recognition. The dataset is an open-access object detection and tracking benchmark collected under ethical principles and released for research use. Thus, this research did not entail direct human subject interaction and did not need further ethical approval.

Consent for Publication Participants.

Consent for publication was given by all participants

Competing Interests

The authors declare no competing interests.

Code Repository:

The python codes for the experiments done in this study can be found at:

https://github.com/Gangeshwar/YOLOv8SR

References

[1] Z. Zhang and L. Zhu, “A review on unmanned aerial vehicle remote sensing: Platforms,

Manuscript Received 05 May 2025; Revised 20 August 2025; Accepted 5 October 2025; Available online 14 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.888

sensors, data processing methods, and applications," *Drones*, vol. 7, 2023, doi: 10.3390/drones7060398.

[2] H. Xu, W. Zheng, F. Liu, P. Li, and R. Wang, "Unmanned aerial vehicle perspective small target recognition algorithm based on improved YOLOv5," *Remote Sens (Basel)*, vol. 15, 2023, doi: 10.3390/rs15143583.

[3] A. Ramachandran and A. K. Sangaiah, "A review on object detection in unmanned aerial vehicle surveillance," *Int. J. Cogn. Comput. Eng.*, vol. 2, pp. 215–228, 2021, doi: 10.1016/j.ijcce.2021.11.005.

[4] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with YOLOv8," *arXiv preprint arXiv:2305.09972*, 2023, doi: 10.48550/arXiv.2305.09972.

[5] Y. Zhou, Y. Z. Yan, H. Y. Chen, and S. H. Pei, "Defect detection of photovoltaic cells based on improved Yolov8," *Laser Optoelectron. Prog.*, pp. 1–17, Oct. 2023.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *arXiv preprint arXiv:1506.02640*, 2015, doi: 10.48550/arXiv.1506.02640.

[7] Z. Luo, C. Wang, Z. Qi, and C. Luo, "LA_YOLOv8s: A lightweight-attention YOLOv8s for oil leakage detection in power transformers," *Alexandria Eng. J.*, vol. 92, pp. 82–91, 2024, doi: 10.1016/j.aej.2024.02.054.

[8] Z. Yu and others, "An enhancement algorithm for head characteristics of caged chickens detection based on cyclic consistent migration neural network," *Poultry Sci.*, vol. 103, 2024, doi: 10.1016/j.psj.2024.103663.

[9] S. Kumar and H. Kumar, "LUNGCOV: A diagnostic framework using machine learning and imaging modality," *Int. J. Tech. Phys. Probl. Eng. (IJTPE)*, vol. 14, 2022.

[10] S. Kumar and H. Kumar, "Classification of COVID-19 X-ray images using transfer learning with visual geometrical groups and novel sequential convolutional neural networks," *MethodsX*, vol. 11, 2023, doi: 10.1016/j.mex.2023.102295.

[11] Z. Diao and others, "Navigation line extraction algorithm for corn spraying robot based on improved YOLOv8s network," *Comput. Electron. Agric.*, vol. 212, 2023, doi: 10.1016/j.compag.2023.108049.

[12] C. Fu and L. Cohen, "Conic Linear Units: Improved Model Fusion and Rotational-Symmetric Generative Model," in *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, SCITEPRESS - Science and Technology Publications, 2024, pp. 686–693. doi: 10.5220/0012406500003660.

[13] Ultralytics Glossary, "SiLU (Sigmoid Linear Unit)," url: https://www.ultralytics.com/glossary/silu-sigmoid-linear-unit#how-silu-works, accessed on 9th Oct 2025.

[14] Ultralytics YOLO Docs, "Home", url: https://docs.ultralytics.com, accessed on 9th Oct 2025.

[15] A. Paul and others, "Smart solutions for capsicum harvesting: Unleashing the power of YOLO for detection, segmentation, growth stage classification, counting, and real-time mobile identification," *Comput. Electron. Agric.*, vol. 219, 2024, doi: 10.1016/j.compag.2024.108832.

[16] A. Wang and others, "NVW-YOLOv8s: An improved YOLOv8s network for real-time detection and segmentation of tomato fruits at different ripeness stages," *Comput. Electron. Agric.*, vol. 219, 2024, doi: 10.1016/j.compag.2024.108833.

[17] W. Yang, X. Zhang, Y. Tian, W. Wang, and J.-H. Xue, "Deep learning for single image super-resolution: A brief review," *IEEE Trans. Multimedia*, pp. 3106–3121, 2019, doi: 10.1109/TMM.2019.2919431.

[18] I. V Grossu, O. Savencu, M. Verga, and N. Verga, "Optimization technique for increasing resolution in computed tomography imaging," *MethodsX*, vol. 10, 2023, doi: 10.1016/j.mex.2023.102228.

[19] R. Rombach and others, "High-resolution image synthesis with latent diffusion models," *arXiv preprint arXiv:2112.10752*, 2021, doi: 10.48550/arXiv.2112.10752.

[20] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual networks of residual networks: Multilevel residual networks for image super-resolution," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 6, pp. 1126–1140, 2019, doi: 10.1109/TCSVT.2018.2858544.

[21] X. Wang *et al.*, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, 2018. doi: 10.48550/arXiv.1809.00219.

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Machine Learning (ICML)*, 2015, pp. 448–456. doi: 10.48550/arXiv.1502.03167.

[23] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Machine Learning (ICML)*, 2010, pp. 807–814.

Manuscript Received 05 May 2025; Revised 20 August 2025; Accepted 5 October 2025; Available online 14 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.888

[24] W. Shi *et al.*, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 1874–1883. doi: 10.1109/CVPR.2016.207.

[25] VisDrone, "The dataset for drone based detection and tracking is released, including both image/video, and annotations," 2020.

[26] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," Jul. 2018.

[27] S. Kumar and others, "A methodical exploration of imaging modalities from dataset to detection through machine learning paradigms in prominent lung disease diagnosis: A review," *BMC Med. Imaging*, vol. 24, 2024, doi: 10.1186/s12880-024-01192-w.

[28] P. Zhu and others, "Detection and tracking meet drones challenge," *arXiv preprint arXiv:2001.06303*, 2020, doi: 10.48550/arXiv.2001.06303.

[29] M. Elham and J. Kim, "Drone-based crowd monitoring using adaptive object detection and tracking," *Sensors*, vol. 21, no. 5, p. 1542, 2021, doi: 10.3390/s21051542.

[30] H. T. Tran, D. T. Nguyen, and N. T. Vo, "Improved UAV image recognition using an augmented YOLOv5s with attention mechanism," *J Intell Robot Syst*, vol. 107, no. 3, p. 47, 2023, doi: 10.1007/s10846-023-01796-3.

[31] M. Mahmood and S. Abbas, "Multiscale detection in UAV surveillance using hybrid YOLOv4-tiny framework," *Electronics (Basel)*, vol. 11, no. 19, p. 3032, 2022, doi: 10.3390/electronics11193032.

[32] Y. Zhou and H. Zhang, "Real-time vehicle detection and tracking in drone footage using transformer-enhanced YOLOv8," *Remote Sens (Basel)*, vol. 16, no. 1, p. 112, 2024, doi: 10.3390/rs16010112.

[33] D. Gupta and S. Rathore, "An improved YOLOv8 for multiclass object detection in aerial imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, p. 5500210, 2023, doi: 10.1109/TGRS.2023.5500210.

[34] L. Wang and H. Li, "Small object classification using YOLOv8-SR in complex urban landscapes," *J Vis Commun Image Represent*, vol. 89, p. 103759, 2023, doi: 10.1016/j.jvcir.2023.103759.

[35] X. Zhao and Y. Sun, "Multi-class object detection in UAV images with attention-based YOLO framework," *Remote Sens (Basel)*, vol. 15, no. 8, p. 1920, 2023, doi: 10.3390/rs15081920.

[36] D. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[37] M. Goyal, R. Singh, and A. K. Sangaiah, "Hyper-parameter tuning and performance evaluation of deep learning models for UAV image classification," *Neural Comput. Appl.*, vol. 34, pp. 11573–11585, 2022, doi: 10.1007/s00521-021-06183-5.

[38] N. S. Raza and others, "A comprehensive evaluation of optimization algorithms for YOLO-based object detectors in UAV applications," *Drones*, vol. 7, no. 9, p. 531, 2023, doi: 10.3390/drones7090531.

[39] X. Yang, L. Nie, Y. Zhang, and L. Zhang, "Image Generation and Super-Resolution Reconstruction of Synthetic Aperture Radar Images Based on an Improved Single-Image Generative Adversarial Network," *Information*, vol. 16, no. 5, p. 370, 2025, doi: 10.3390/info16050370.

[40] A. Rouhbakhshmeghrazi and J. Li, "Super-Resolution Reconstruction of UAV Images with GANs: Achievements and Challenges," *Remote Sens (Basel)*, vol. 14, no. 1, p. 121, 2022, doi: 10.3390/rs14010121.

[41] J. Wang, H. Li, and Z. Gao, "A No-Reference Quality Assessment Metric Based on PSNR and SSIM Learning Fusion for Reconstructed Images," *IEEE Access*, vol. 9, pp. 143562–143573, 2021, doi: 10.1109/ACCESS.2021.3120765.

[42] D. Liu, J. Yang, C. Huang, and J. Huang, "Image quality assessment for super-resolution: A survey," *Neurocomputing*, vol. 469, pp. 427–446, 2022, doi: 10.1016/j.neucom.2021.10.087.

[43] H. Lu, Y. Zhang, and J. Liu, "Objective image quality assessment based on improved PSNR and SSIM for super-resolution evaluation," *Signal Process. Image Commun.*, vol. 103, 2022, doi: 10.1016/j.image.2022.116696.

[44] M. Zhou, F. Zhu, and Y. Yang, "Perceptual image quality assessment based on deep feature similarity and multi-scale SSIM," *Multimedia Tools Appl.*, vol. 81, pp. 28521–28542, 2022, doi: 10.1007/s11042-021-11432-5.

[45] A. Tripathi, V. Gohokar, and R. Kute, "Comparative Analysis of YOLOv8 and YOLOv9 Models for Real-Time Plant Disease Detection in Hydroponics," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 17269–17275, Oct. 2024, doi: 10.48084/etasr.8301.

[46] P. Hidayatullah, N. Syakrani, M. R. Sholahuddin, T. Gelar, and R. Tubagus, "YOLOv8 to YOLO11: A Comprehensive Architecture In-depth Comparative Review," Apr. 2025.

Manuscript Received 05 May 2025; Revised 20 August 2025; Accepted 5 October 2025; Available online 14 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.888

Author's Biography

Gangeshwar Mishra is a seasoned technology leader and Director at App2Mobile, with over 15 years of experience in driving innovation through Artificial Intelligence (AI). He began his journey in the field of image processing and has built a diverse portfolio across e-commerce, cybersecurity, and education technologies. An alumnus of B.Tech (2011) and M.Tech (2015), Gangeshwar has held key roles in several esteemed organizations, including the Times Group, where he led high-impact, scalable AI initiatives. His expertise lies in architecting intelligent systems that process high volumes of data and deliver tangible, real-world impact. Recognized for his visionary leadership, deep technical acumen, and ethical approach to AI, he has contributed to nationally and internationally acclaimed projects. Passionate about transforming ideas into intelligent solutions, Gangeshwar continues to shape the AI landscape through strategic innovation, mentorship, and thought leadership.

Dr. Rohit Tanwar is an accomplished academician and researcher with over 14 years of teaching experience in the field of Computer Science and Engineering. Currently serving as an Associate Professor at Shri Mata Vaishno Devi University, Katra (J&K), he has held various academic positions, including at the University of Petroleum and Energy Studies, Dehradun, and Manav Rachna University, Faridabad. He earned his Ph.D. in Computer Science and Engineering from Kurukshetra University, with earlier degrees in Computer Engineering from YMCA University of Science and Technology and U.I.E.T. Kurukshetra. Dr. Tanwar has made substantial contributions to research in cybersecurity, steganography, IoT, and AI-enabled healthcare systems. He holds multiple Indian patents and copyrights, and has authored and edited numerous Scopus-indexed books and research papers published by reputed publishers such as Elsevier, Springer, Taylor & Francis, and MDPI. He has supervised Ph.D. and M.Tech dissertations, served as a reviewer and editor for prominent international journals, and been a session chair and keynote speaker at multiple national and international conferences. His research excellence has been recognized with awards, including the AOTA Research Excellence Award (2023) and research rewards at UPES. In addition to his academic roles, Dr. Tanwar has also taken on administrative responsibilities, including roles in academic affairs, research coordination, and institutional accreditations. He is a Senior IEEE Member and actively engaged in curriculum development, faculty training, and promoting innovation in computer science education.

Dr. Prinima Gupta is a highly accomplished academic and researcher serving as a Professor in the Department of Computer Science & Technology and Director of the Doctoral Program at Manav Rachna University, Faridabad. A Ph.D. graduate (2013) specializing in Ad hoc networks, she brings over 18 years of experience in teaching, research, and academic leadership. Dr. Gupta's expertise spans Information Security, Data Mining, ad hoc networks, and computer graphics. She has guided 8 Ph.D. candidates (four awarded, four in progress), and mentored numerous M.Tech/ UG students. A prolific author, she has published over 45 peer-reviewed articles, including significant work in video/image steganography, temporal data forecasting, and disaster prediction using big data analytics. Her research has shaped intelligent, secure computing across multiple applications. Beyond research, Dr. Gupta plays integral roles as Director of the Doctoral Program, PG Coordinator, and VAC Coordinator in her department. She is an active reviewer for journals and conferences and a lifetime member of the Indian Society for Technical Education. Additionally, she has chaired sessions at international forums such as ICMLDE 2023. Her multifaceted leadership reflects a commitment to advancing secure, data-driven technology. Through strategic academic stewardship and pioneering research, Dr. Gupta is a respected authority in AI-enabled computing and cybersecurity, shaping future engineers and knowledge frontiers with distinction.

Manuscript Received 05 May 2025; Revised 20 August 2025; Accepted 5 October 2025; Available online 14 October 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i4.888