Applied Computer Science 2026, vol. 22, no. 2, pp. 33–47
https://doi.org/10.35784/acs_9205

Submitted: 2026-02-13 | Revised: 2026-05-05 | Accepted: 2026-05-14

Keywords: UAV, high-voltage power grid, object detection, backbone

Thi Thanh Tan NGUYEN [1], *Thi Thu Nga VU* [1*]
[1] Electric Power University, Vietnam, tanntt@epu.edu.vn, ngavtt@epu.edu.vn
[*] Corresponding author: ngavtt@epu.edu.vn

Assessing the effectiveness of one-stage and two-stage methods for identifying high-voltage power grid equipment in UAV imagery

Abstract

Unmanned Aerial Vehicle (UAV) imagery, augmented by advanced deep learning architectures, has become an integral approach to the automated inspection and structural monitoring of high-voltage power grids. This study evaluates the practical applicability and speed-accuracy trade-offs of single-stage versus two-stage object detection models for identifying critical power grid components. Specifically, two highly optimized networks were developed and empirically validated: HVE-YOLO11, which uses the latest YOLO11 architecture enhanced with spatial attention mechanisms, and HVE-MASK-R-CNN, which uses a rigorous ResNet-101 Feature Pyramid Network (FPN) backbone. Leveraging a newly curated, diverse dataset of 51,800 augmented images capturing six distinct equipment classes under fluctuating meteorological conditions, the models were evaluated for Mean Average Precision (mAP) and computational throughput measured in Frames Per Second (FPS). Empirical results demonstrate that the single-stage HVE-YOLO11 shatters the traditional speed-accuracy dichotomy, significantly outperforming the two-stage model in both inference velocity (162.9 FPS versus 75.8 FPS in intensive benchmarking) and spatial accuracy (an mAP@0.5 of 0.972 compared to 0.855). These findings provide actionable, highly quantified benchmarks for deploying real-time, AI-driven diagnostic systems on hardware-constrained edge-computing UAV platforms.

1. INTRODUCTION

High-voltage power transmission lines are crucial for a stable electricity supply but are vulnerable to weather-related damage, which affects grid reliability. Regular inspections are essential to prevent outages, ensure safety, comply with regulations, and reduce maintenance costs. UAV technology has become an effective tool for monitoring these systems, particularly for detecting components on high-voltage lines (Santos et al., 2024). Object detection has shifted from computationally expensive methods like SIFT and HOG to deep learning approaches. These can be classified into two types: two-stage and one-stage detectors. Two-stage detectors first generate proposals and then classify them, while one-stage detectors perform both steps in a single pass, making them faster and more efficient for real-time applications (Faisal et al., 2025). However, two-stage methods offer higher accuracy by reducing detection errors but are slower. YOLOv11, a one-stage model, excels at detecting small objects, while Mask R-CNN, a two-stage model, is known for its accuracy (Wu et al., 2020).

This paper focuses on improvements to the YOLO11 and Mask R-CNN architectures. Based on these advancements, we developed the HVE-YOLO11 and HVE-MASK-R-CNN models for detecting six types of equipment on high-voltage power grids: electric cables (elec-cable), silicon insulators (slc-insulator), glass insulators (gl-insulator), steel poles (steel-pole), monopoles (monopole), and vibration dampers (vib-damper).

Despite the rapid proliferation of deep learning frameworks within the domain of power grid inspections, a critical research gap persists regarding the optimal deployment of next-generation, attention-augmented single-stage architectures in highly occluded, real-world operational environments. While existing literature has thoroughly validated preceding models such as YOLOv8 and traditional Mask R-CNN frameworks, the recent architectural introduction of YOLO11—which natively integrates computationally efficient C3k2 blocks alongside novel Cross-Stage Partial Spatial Attention (C2PSA) modules—has not yet been empirically evaluated against state-of-the-art two-stage detectors within this specialized infrastructure domain. Furthermore, there remains a pronounced scarcity of literature that rigorously quantifies the speed-accuracy

trade-offs of these contrasting architectures when processing high-resolution (1820x720) UAV imagery captured under diverse, extreme meteorological conditions. This study directly addresses this empirical void by providing a definitive, quantified benchmark that compares the architectural efficacy, theoretical computational overhead, and real-time inference capabilities of the single-stage HVE-YOLO11 against the two-stage HVE-MASK-R-CNN model.

Main contributions in our research:

We have created a large dataset by capturing UAV images of the high-voltage powerline grid in Vietnam under various natural conditions, including mild and strong sunlight, shade, drizzle, and light to normal winds, at different distances from the UAV to the equipment. This diverse dataset is valuable for both research and practical applications, enhancing algorithm accuracy and stability.

We developed two models for identifying equipment on high-voltage power grids, based on the YOLO11 and Mask R-CNN architectures, which represent the top network architectures for single-stage and two-stage object detection, respectively. A comprehensive comparison of their performance in high-voltage grid equipment recognition was conducted, setting benchmarks for future AI-based solutions in power grid inspection.

The paper is structured as follows: section 2 covers related approaches, section 3 details the proposed method, section 4 describes testing and evaluation, and section 5 discusses conclusions and future directions.

2. RELATED WORK

Vemula et al. (2020) used Mask R-CNN to detect electric poles and components such as insulators and transformers. Chen et al. (2020) proposed a method for detecting foreign objects on high-voltage power lines using Mask R-CNN to overcome the limitations of traditional methods, which are affected by weather and environmental factors. Their study confirmed that Mask R-CNN outperforms in terms of speed, efficiency, and accuracy, thereby enhancing grid reliability. Zhou et al. (2022) introduced ARG-Mask R-CNN, an improved deep learning network that combines Attention, Rotation, and Genetic Algorithms for insulator fault detection in infrared images. In Chen et al. (2024), the authors integrated contextual information into the Mask R-CNN model to address occlusion and small object sizes in power line component detection. In Tang et al. (2025), an improved Mask R-CNN model integrated Reparameterized Convolution (RefConv) and Efficient Channel Attention (ECA) for better feature extraction, multi-scale fusion, and noise reduction. Wang et al. (2023) proposed an enhanced YOLOv8m model for detecting foreign objects on power lines, incorporating a Global Attention Module (GAM) and Focal-EIoU loss function to address occlusion and sample imbalance. In Shao et al. (2024), the authors introduced the TL-Yolo model based on YOLOv8, enhancing feature extraction, fusion, and sample balance. Rong et al. (2025) proposed a real-time monitoring model using YOLOv8 with directional filters for more accurate power line detection. In Z. Wang et al. (2025), an improved Line-YOLO algorithm based on YOLOv8s-seg addressed tilt angle detection using DCNv4 deformable convolution and EMA attention mechanisms. In C. Wang et al. (2025), the authors enhanced the YOLOv7 model for insulator defect detection, using the RFB module, CA mechanism, and WIoU loss function to improve feature extraction and training on low-quality samples. Ji et al. (2025) introduced an improved YOLOv11 algorithm for insulator defect detection on distribution lines, excelling in complex environments. Finally, in Xu et al. (2025), a deep learning-based model derived from YOLOv8 was proposed to detect transmission line insulator faults, integrating advanced modules like C3k2 and C2fCIB.

3. HIGH-VOLTAGE POWER GRID EQUIPMENT IDENTIFICATION MODELS

3.1. HVE-YOLO11 architecture

The HVE-YOLO11 model is built on the latest YOLO11 architecture, optimized for edge-device deployment and rapid real-time inference. As shown in the architectural schematic (Fig. 1), the network is divided into three sequential operational phases: Part 1: The Backbone, Part 2: The Neck (Feature Aggregation), and Part 3: The Detection Head.

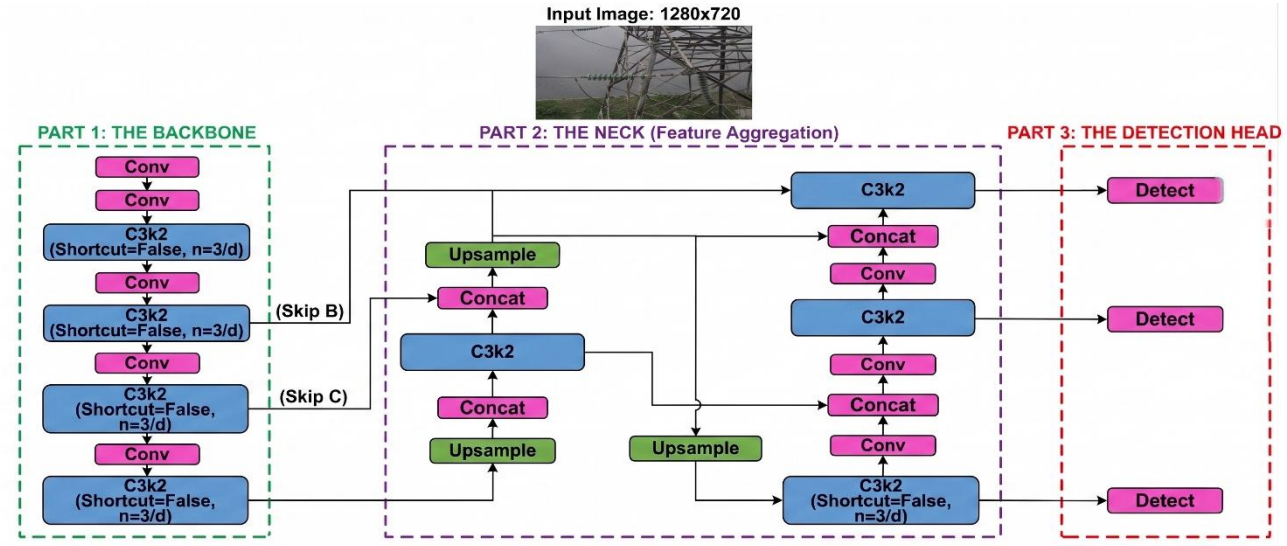


Fig. 1. HVE-YOLO11 architecture

Part 1, the Backbone, functions as the primary multi-scale feature extractor. It is composed of alternating Convolutional (Conv) layers and computationally efficient C3k2 blocks (configured with $Shortcut = False$ and $n = 3/d$). The C3k2 modules accelerate spatial processing while extracting deep semantic representations from the 1280×720 input tensor. Crucially, intermediate semantic maps are explicitly preserved and routed to the subsequent aggregation stages via specific skip connections, denoted as Skip B and Skip C.

Part 2, the Neck, is responsible for complex feature aggregation. It utilizes Upsample operations and Concatenation (Concat) modules to seamlessly fuse the high-resolution features from Skip B and Skip C with deeper, semantically rich layers. This top-down and bottom-up pathway is interspersed with additional Conv and C3k2 blocks, ensuring robust multi-scale contextual awareness.

Finally, Part 3, the Detection Head, consists of three parallel 'Detect' modules that process the aggregated features to predict bounding box coordinates and classification probabilities across three distinct scales. To further elevate the scientific rigor and localization precision of the model, the entire architecture undergoes end-to-end optimization utilizing a composite multi-task loss function. The total loss ($\mathcal{L}_{total}$) mathematically integrates three distinct optimization metrics:

$$\mathcal{L}_{total} = \lambda_{box}\mathcal{L}_{CIoU} + \lambda_{cls}\mathcal{L}_{BCE} + \lambda_{dfl}\mathcal{L}_{DFL} \quad (1)$$

Here, $\mathcal{L}_{CIoU}$ represents the Complete Intersection over Union loss, which evaluates the exact geometric overlap, aspect ratio, and center-point distance for precise bounding box regression. $\mathcal{L}_{BCE}$ denotes the Binary Cross-Entropy loss governing categorical classification accuracy, and $\mathcal{L}_{DFL}$ is the Distribution Focal Loss, which refines bounding box boundary predictions by modeling spatial coordinates as continuous probability distributions. The coefficients (λ_{box}, λ_{cls}, λ_{dfl}) act as empirically tuned weighting factors to balance the optimization process.

3.2. HVE-MASK-R-CNN architecture

The HVE-MASK-R-CNN model represents a highly robust, two-stage instance segmentation methodology. As detailed in the structural diagram (Fig. 2), the network's predictive pipeline is systematically divided into four integrated modules.

Part 1: The Backbone leverages a deep ResNet-101 architecture seamlessly unified with a Feature Pyramid Network (FPN). This structure establishes a top-down pathway augmented by lateral connections, enabling the extraction of multi-scale, semantically robust 'Feature Maps' from the raw imagery. Part 2: The Region Proposal Network (RPN) operates dynamically on these continuous Feature Maps. It applies a 3×3 Convolution, branching into two parallel 1×1 Convolutional pathways: one utilizing a Softmax activation to evaluate objectness scores, and another performing Bounding Box Regression (Bbox Reg). Together, these branches generate discrete target 'Proposals'.

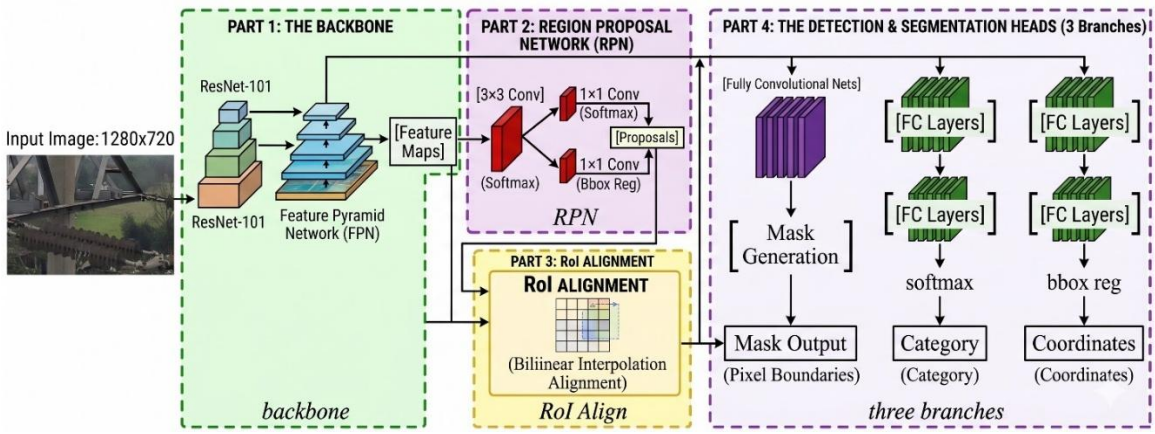


Fig. 2. HVE-MASK-R-CNN architecture

Part 3 introduces RoI Alignment to address the critical need for spatial quantization precision. Rather than employing traditional integer-based pooling, this module utilizes Bilinear Interpolation Alignment to precisely map the continuous geometric coordinates of the generated Proposals onto the discrete Feature Maps. The mathematical interpolation for a sampled spatial point (x, y) within the RoI is calculated as:

$$f(x,y) = \sum_{i,j\in\{0,1\}} (1-|x-x_i|)(1-|y-y_j|)f(x_i,y_j) \tag{2}$$

where $f(x_i, y_j)$ represents the continuous feature values computed directly from the four nearest integer pixel coordinates, guaranteeing exact pixel-to-pixel localization boundaries.

Finally, Part 4: The Detection & Segmentation Heads process the aligned RoI tensors through three parallel task branches. One branch uses Fully Convolutional Networks (FCNs) dedicated to 'Mask Generation', producing precise 'Pixel Boundaries'. Concurrently, two separate Fully Connected (FC) layer branches yield the final 'Category' (via Softmax) and 'Coordinates' (via Bbox Reg). The entire framework is jointly optimized using a comprehensive loss function that penalizes all three predictive deviations simultaneously:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{box} + \mathcal{L}_{mask} \tag{3}$$

where $\mathcal{L}_{mask}$ computes the average binary cross-entropy loss evaluated independently for each class, preventing spatial competition between classes during the segmentation phase.

3.3. Target equipment classes and morphological characteristics

In this context, the YOLOv11 and Mask R-CNN network architectures will be used to detect and classify six basic types of equipment (objects) in high-voltage power grids, including: monopoles, steel lattice towers, electrical conductors, vibration dampers, glass insulators, and silicon insulators.

High-voltage power grid poles are typically categorized into three types: concrete poles, monopoles, and steel lattice towers (Fig. 3). Monopoles, made from reinforced concrete or metal, consist of a single pole body without additional structural components like steel lattice. These poles, usually cylindrical or conical in shape, are commonly used for low-intensity power lines or in areas with simpler structural needs. Steel lattice towers are constructed from steel beams shaped like H, I, or other specialized structures. These poles are often reinforced with horizontal bars, diagonal braces, or other components to enhance stability.

Fig. 3. High-voltage power poles

Electric wires (Fig. 4) guide the path of electric current (electrical conduction). The materials used to manufacture conductors can be aluminum, copper, steel, or aluminum alloys. The conductors on high-voltage transmission grids are made in two types: multi-strand wires and hollow conductors (Fig. 4). Multi-strand wire consists of many strands twisted together, with the number of strands depending on the cross-sectional area, typically 7, 19, 37, or 61 strands twisted in opposite directions to prevent the wire from unraveling. Hollow conductors are designed to meet the requirement of large wire diameters to reduce the electric corona and minimize energy loss.

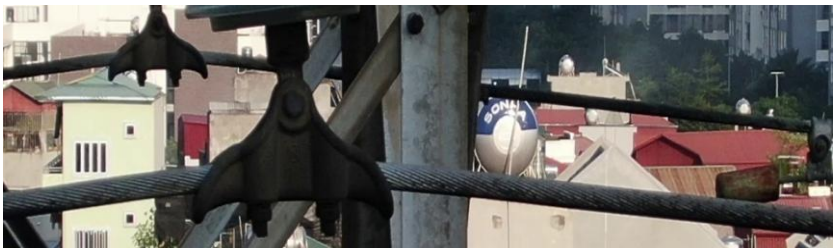


Fig. 4. Electric wire

Insulators are devices in electrical networks that maintain a safe distance between power lines and structures or between the lines themselves. They help ensure safety by reducing the risk of fire and explosion. Common insulating materials include ceramics, glass, composites, and silicon. Insulators are classified into three types: vertical, suspended, and through insulators. Suspended insulators are typically made of multiple insulator bowls linked in a chain, with the number depending on voltage and operating conditions (Fig. 5).

Fig. 5. Insulator

The vibration damper primarily reduces the impact of bending stress on the transmission line and minimizes the overhead cable's vibration amplitude. The anti-vibration weight consists of two counterweights connected by steel cables and suspended from the conductor using suspension clamps (Fig. 6).

Fig. 6. Vibration damper

To meet the research objectives, we developed two models for recognizing six types of equipment on high-voltage power grids, utilizing the YOLOv11 and Mask R-CNN network architectures. The first model is referred to as HVE-YOLO11 (High Voltage Equipment Recognition based on YOLOv11 architecture), while the second model is HVE-MASK-R-CNN (High Voltage Equipment Recognition based on Mask R-CNN architecture).

Fig. 7 delineates the comprehensive computational pipeline utilized throughout this study. Raw high-resolution images acquired via UAV sensors undergo standardized preprocessing and normalization to a 1280×720 spatial resolution. Following orthographical bounding-box annotation in standard COCO JSON format, sophisticated data augmentation techniques are applied to mitigate inherent class imbalances and enhance environmental diversity. The subsequent algorithmic phases involve initializing the selected architectural backbones with pre-trained weights, conducting iterative network optimization, and executing a rigorous multi-metric performance evaluation on the validation corpus.

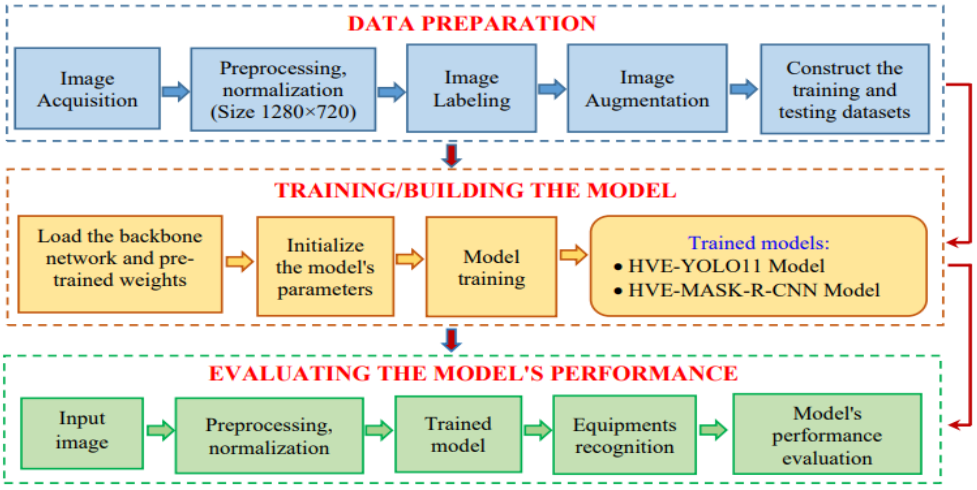


Fig. 7. The diagram of building equipment recognition models for high-voltage power grids

The choice of backbone and pre-training weights significantly affects the network's performance, including speed, computational cost (FLOP), and accuracy in object detection and identification. The core function of a backbone in an object detection network is to extract features. This network takes a raw input image and produces a set of multi-scale, semantically rich feature maps. This process is not a single transformation but a sequence of structured processing steps.

In our study, for the HVE-YOLOv11 architecture, the backbone is designed as a deep stack of Convolutional Layers (Conv) combined specifically with C3k2 blocks to generate feature maps at varying resolutions. As shown in the schematic, this architectural variation relies heavily on the C3k2 block—a computationally efficient implementation of the Bottleneck Cross Stage Partial (CSP) framework—to preserve gradient routing while maximizing structural efficiency during downsampling.

The backbone of the HVE-MASK-R-CNN architecture is a combination of the ResNet101 network and the Feature Pyramid Network (FPN). ResNet101 is a residual network with 101 layers, used to extract features during object recognition. The ResNet network utilizes residual blocks to mitigate signal degradation during the training of deep networks. FPN (Feature Pyramid Network) is an improvement that enables the network to learn features at different levels of the image (low and high layers). This helps improve the detection of objects at various sizes, especially small or large objects. FPN uses features from ResNet101 at different levels and builds a system of higher-resolution features in the upper layers of the network, while the deeper layers capture larger features from the image. The combination of ResNet101 and FPN creates a powerful architecture for detecting objects at various resolutions.

The final stage involves testing/using the model that was trained in the previous step. From any given input image, it is first normalized to a size of 1280×720, and then the model's effectiveness is evaluated based on the metrics presented in the next section.

4. EXPERIMENT RESULTS

4.1. Experimental environment

The authors used the Python 3 environment to implement the algorithm and experimental model. The test machine configuration includes an Intel Core i7-9700 CPU with a 4.7 GHz (Max Turbo Frequency), 32 GB RAM. The computer is equipped with two 12 GB GDDR5 Nvidia Tesla K80 graphics cards.

4.2. Performance evaluation metrics

To evaluate the effectiveness of the method, we use several metrics, including the Confusion Matrix, Precision, Recall, F1-Score, F1-Confidence curve, Precision-Confidence curve, Precision-Recall curve, Recall-Confidence curve, AP, and mAP. In multi-class classification problems, the confusion matrix is a fundamental tool for evaluating classification model performance. For multi-class classification, the confusion

matrix is extended to an N×N matrix, where N is the number of classes. The structure of a confusion matrix includes:

- C[i][j]: The number of instances where the actual class is i, but the model predicts class j.
- Along the diagonal (C[i][i]): Represents the number of correct predictions for class iii.
- The off-diagonal elements (i≠j): Represent misclassifications.
- True Positive (TP): The number of instances where the model correctly predicts the class for instances that belong to that class.
- True Negative (TN): The number of instances where the model incorrectly predicts a sample as class i, while it actually belongs to a different class.
- False Negative (FN): The number of instances where the model fails to correctly classify a sample belonging to class iii (it predicts another class).
- True Negative (TN): The number of instances where the model correctly predicts that a sample does not belong to class i.
- *Accuracy*: This is the ratio of correct predictions to the total number of samples.

$$Accuracy = \frac{\sum_{i=1}^{N} C[i][i]}{\Gamma\text{otal\_number\_of\_samples}} \tag{4}$$

- *Precision* (for each class): This is the ratio of correct predictions of class i to the total number of predictions as class i (both correct and incorrect):

$$\mathcal{P}recision(i) = \frac{C[i][i]}{\sum_{j=1}^{N} C[j][i]} \tag{5}$$

- *Recall* (for each class): This is the ratio of actual class i samples that the model correctly predicts.

$$Recall = \frac{T_P}{T_P + F_N} \tag{6}$$

- *F1-Score*: It is a combination of Precision and Recall, particularly useful when considering both metrics:

$$F1(i) = 2 \times \frac{\text{Precision}(i) \times \text{Recall}(i)}{\text{Precision}(i) + \text{Recall}(i)} \tag{7}$$

- *F1-Confidence Curve* is generated by adjusting the confidence thresholds and calculating the corresponding F1-Scores for each threshold value. It combines both precision and recall, offering a comprehensive view of the model's performance in detecting objects within images.
- *The Precision-Confidence Curve* is a graph that illustrates how precision changes across different confidence thresholds. It highlights the variation in precision as the confidence level of predictions is adjusted, helping you understand the trade-off between precision and the confidence threshold in object recognition tasks.
- The *Precision-Recall Curve* is a powerful tool for evaluating the performance of object detection models, especially in tasks where the positive class has significantly fewer samples than the negative class. This metric helps analyze the relationship between ***precision*** and ***recall*** as the prediction threshold (confidence threshold) is adjusted.
- *The Recall-Confidence Curve* is a crucial tool for assessing the performance of object detection models, particularly in analyzing the relationship between recall and confidence threshold as the confidence levels of the model's predictions are adjusted. It plots recall (on the y-axis) against the confidence threshold (on the x-axis) at various threshold values. This metric provides a clearer understanding of the trade-off between recall and confidence threshold in object detection tasks.
- *Average Precision (AP)* is a metric used to evaluate the performance of an object detection model by combining Precision and Recall in the classification process. AP is calculated from the Precision-Recall curve, which illustrates how Precision and Recall change as the confidence threshold varies. AP represents the area under the Precision-Recall curve (Area Under Curve - AUC). It offers a

comprehensive assessment of the model's ability to accurately predict objects across varying confidence thresholds.

– *Mean Average Precision (mAP)* is a composite metric used to measure the performance of an object detection model across multiple classes (categories). mAP is the average of the AP values for each class:

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i \tag{8}$$

Where: N – the number of object classes, and AP_i is the Average Precision of class i.

- *Inference Speed*: Inference *speed* is typically measured in milliseconds (ms) or frames per second (FPS).

4.3. Experimental data

The effectiveness of the models is evaluated on a dataset collected from the 110 kV high-voltage grid. Data collection was conducted using the DJI Mavic 2 Pro drone. In this study, we focus on identifying power poles, transmission lines, and equipment mounted on them. Structurally, high-voltage transmission lines are usually arranged in 3- or 6-layer configurations. During the data collection process, the drone was set to automatic flight mode, first flying along the line to capture images of the transmission lines and vibration-damping towers, then flying from top to bottom and left to right at each pole position to capture images of the power poles and the equipment on the poles (insulator chains, insulator accessories). The distance from the drone to the transmission line/pole was set between 4m and 5m. Data collection was performed at different times of the day (morning, noon, evening) and during various weather conditions, including cloudy, drizzle, mild sunshine, strong sunshine, light wind, and moderate wind (wind levels ranging from 1 to 5).

The raw data collected will first be pre-processed to remove irrelevant data, duplicate data, align, remove noise-causing objects, and enhance the quality of the input images. It will then be annotated in the Coco file format. Each annotation data is stored in a file in JSON format, which includes the following information:

```
{
    "info": info,
    "licenses":[licenses],
    "categories":[category],
    "images":[image],
    "annotations":[annotation]
}
```

Where info contains metadata about the dataset, licenses provide details about the dataset's licensing; categories list the names of the objects in the dataset, images store information about the images, including the image path, name, and a unique image ID for each image; annotations contain information about the bounding boxes or segmentation details for the objects in the images (see Fig. 8).

Fig. 8. Image labeling to create the ground truth dataset

The experimental dataset for this study consists of 51800 images. It was compiled from two sources: an initial collection of 30,500 real-world images captured from the transmission grid, and an additional 21300 images generated through data augmentation techniques. To ensure a balanced and robust dataset for training, the original set of images was augmented using both advanced and standard methods. This included the use of generative models and GAN-based image synthesis, as well as common image transformations such as RandomScale, Downscale, GaussNoise, Sharpen, and various flips and rotations. This approach aimed to create a diverse and consistent dataset that complements the real-world data. In total, 69955 object instances

were annotated across this image collection, distributed among six distinct classes. Tab.1 provides a detailed breakdown of these object instances and their distribution across the training and validation sets.

Tab. 1. Experiment data

Class ID	Class name	Training samples	Validation samples	Total samples
0	elec-cable	16643	4851	21494
1	slc-insulator	10316	3702	14018
2	gl-insulator	7343	1825	9168
3	steel-pole	5074	1261	6335
4	monopole	2935	1361	4296
5	vib-damper	11473	3171	14644

The distribution of object classes in the experimental data is shown in Fig. 9 (a). Fig. 9 (b) visualizes the bounding boxes, highlighting the variation in aspect ratios and object sizes. These boxes are drawn around the objects, representing either the entire object or portions of it. The 2D heatmap in Fig. 9 (c) illustrates the distribution of the center positions (x, y) of all objects in the dataset. The x-axis and y-axis represent the relative coordinates of each object's center within the image, ranging from 0 to 1. The coordinate (0,0) corresponds to the top-left corner, while (1,1) corresponds to the bottom-right corner.

The color indicates density: darker areas (dark blue) represent locations with a higher concentration of object centers. From Fig. 9(c), we observe a prominent vertical bright streak at x = 0.5 and a horizontal bright streak at y = 0.5. The intersection point at (0.5, 0.5) is the darkest, highlighting a significant trend: most objects are positioned near the center of the image. This is a common characteristic in many datasets, as photographers often place the main subject in the middle of the frame.

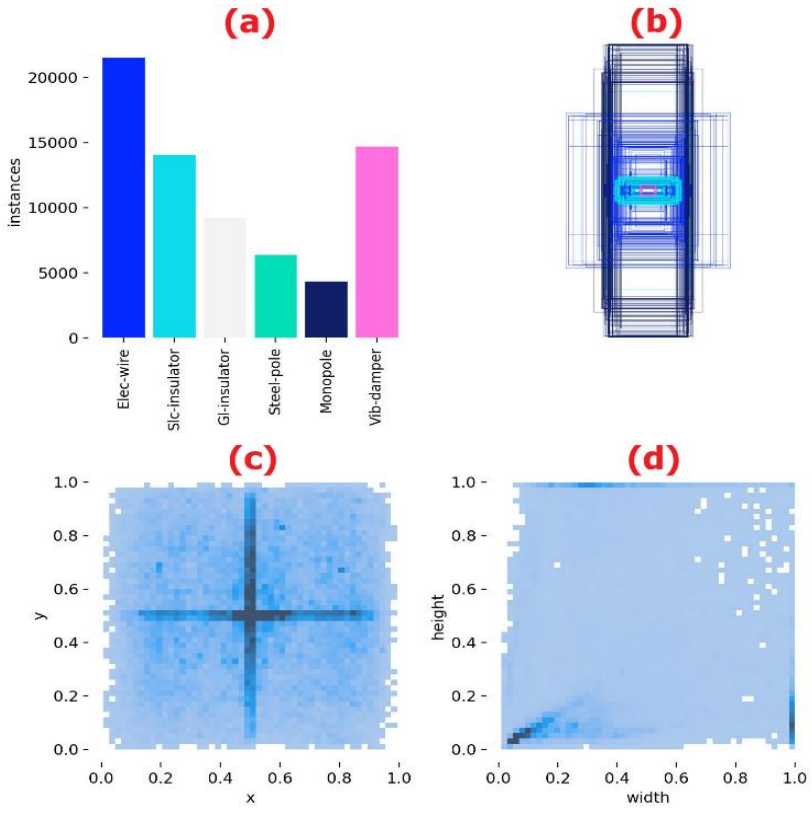


Fig. 9. The distribution of object classes

The 2D spatial heatmap visualizes a pronounced geometric bias, with object bounding box centers heavily clustered at the exact (0.5,0.5) image coordinate. This centralization poses a severe risk of spatial overfitting; convolutional networks trained on such distributions may inadvertently develop an artificial spatial prior, severely penalizing detection confidence when autonomous UAV flight turbulence forces infrastructure components into the extreme peripheral edges of the camera frame. To actively neutralize this algorithmic

vulnerability, our data augmentation pipeline relied heavily upon aggressive geometric transformations. By introducing extreme random cropping, off-axis multi-directional translations, and severe rotational shifts into the training matrix, we artificially forced the target features away from the center of the coordinate system. This augmentation strategy ensures that both the HVE-YOLO11 and HVE-MASK-R-CNN architectures learn strictly scale-invariant and position-invariant morphological features, preserving high-fidelity detection accuracy regardless of the object's absolute location within the real-world 1280×720 operational view.

4.4. Experimental results

The model's effectiveness is evaluated on the experimental dataset using the metrics outlined in Section 4.2. The experiments are conducted on the same test dataset. In this study, the models' performance is analyzed and evaluated based on both accuracy and computational processing efficiency using real-world data.

Regarding model accuracy, the confusion matrices for both HVE-MASK-R-CNN and HVE-YOLO11 (Fig. 10) demonstrate their effectiveness in recognizing six types of equipment in high-voltage power grids: HVE-MASK-R-CNN achieves the highest accuracy for monopoles (99.58%) and steel poles (98.57%). Other accuracies include elec-cable (90.37%), slc-insulator (95.92%), gl-insulator (96.99%), and vib-damper (96.15%). However, elec-cable has a 9.52% misclassification rate. HVE-YOLO11 shows slightly better overall performance, with monopole at 99.56% and steel-pole at 99.37%. Other accuracies include elec-cable (93.59%), slc-insulator (96.03%), gl-insulator (97.59%), and vib-damper (97.79%). Misclassifications are low, especially for elec-cable (6.37%) and slc-insulator (3.97%). While HVE-YOLO11 has higher overall accuracy, particularly for elec-cable and monopole, HVE-MASK-R-CNN offers more consistent results, especially for monopole and steel-pole, with fewer misclassifications. Both models are highly effective, with HVE-YOLO11 excelling in accuracy and HVE-MASK-R-CNN in consistency.

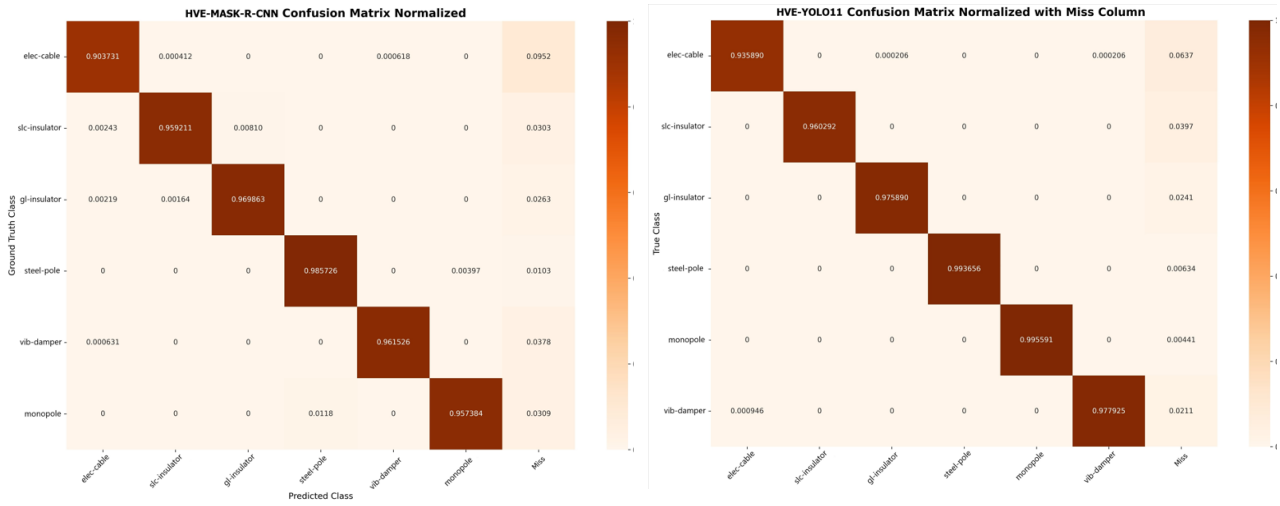


Fig. 10. The model confusion matrices on the experimental dataset

Based on the F1-Confidence curve analysis (Fig. 11), HVE-YOLO11 outperforms HVE-MASK-R-CNN in overall performance, stability, and per-class identification of six equipment types on high-voltage power grids. HVE-YOLO11 achieves a peak F1-Score of 0.95 at a confidence threshold of 0.647, demonstrating an excellent balance between Precision and Recall. In contrast, HVE-MASK-R-CNN reaches a maximum F1-Score of 0.851 at a higher confidence threshold of 0.830, indicating lower performance. The HVE-YOLO11 model is also more flexible, as it maintains high performance across a wide range of confidence thresholds, from 0.25 to 0.9. On the other hand, HVE-MASK-R-CNN's performance is more sensitive to threshold adjustments, making it harder to optimize for real-world applications.

In terms of individual class performance, HVE-YOLO11 consistently achieves high F1-scores across all six classes, with F1-scores above 0.95 for most, including glass insulators, steel poles, monopoles, and vibration dampers. Even for the challenging electric cable class, it achieves an F1-Score of 0.88. In comparison, HVE-MASK-R-CNN excels in some classes but struggles with others, particularly the electric cable class, which scores only 0.66, and the vibration damper class, where performance drops as the confidence threshold

increases. The severe performance degradation of HVE-MASK-R-CNN on the 'elec-cable' class warrants deeper algorithmic interpretation.

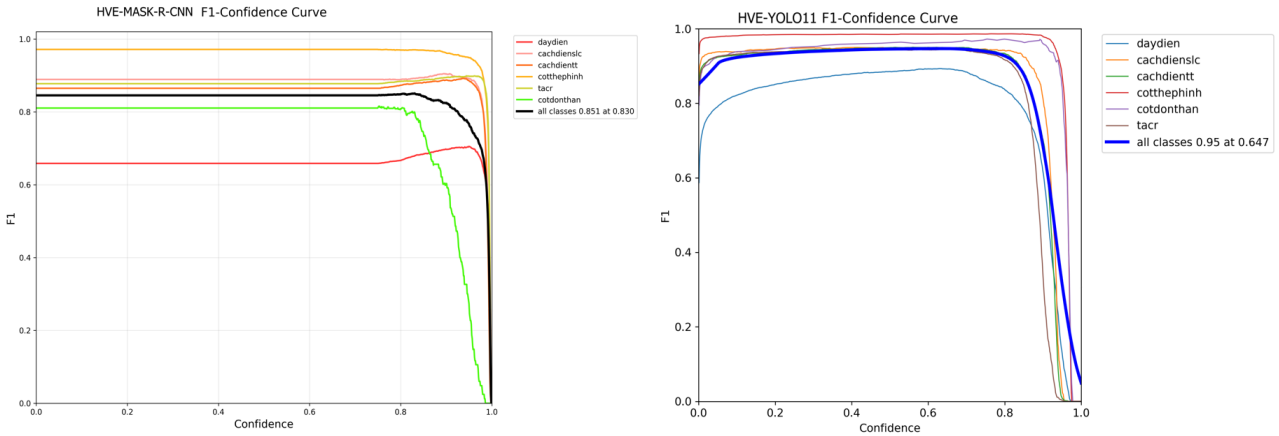


Fig. 11. F1-Confidence curve analysis

Electrical cables present significant detection challenges due to their extreme length, thin structure, and diagonal continuity across the image frame. HVE-MASK-R-CNN relies on RoIAlign to extract localized features from discrete region proposals, making it ineffective for objects with extreme aspect ratios. In such cases, the RPN either generates oversized anchor boxes containing excessive background noise or fragmented proposals that fail to preserve cable continuity. In contrast, HVE-YOLO11 employs a globally aware single-stage convolutional architecture. The integration of the C2PSA module enables self-attention across the entire feature map, allowing the model to learn long-range spatial continuity throughout the 1280×720 image tensor. As a result, YOLO11 effectively overcomes the localized limitations of proposal-based detection and achieves more reliable cable recognition.

The Precision-Confidence curve analysis (Fig. 12) reveals a significant performance difference between the HVE-MASK-R-CNN and HVE-YOLO11 models. HVE-YOLO11 outperforms, showing a strong correlation between confidence and precision. Precision rapidly rises to over 90% at a threshold of 0.2, approaching 100% as confidence increases. In contrast, HVE-MASK-R-CNN's precision remains around 81% across a wide confidence range (0.0 to 0.75), improving only when the threshold exceeds 0.8, with minimal effect on false positives at lower thresholds.

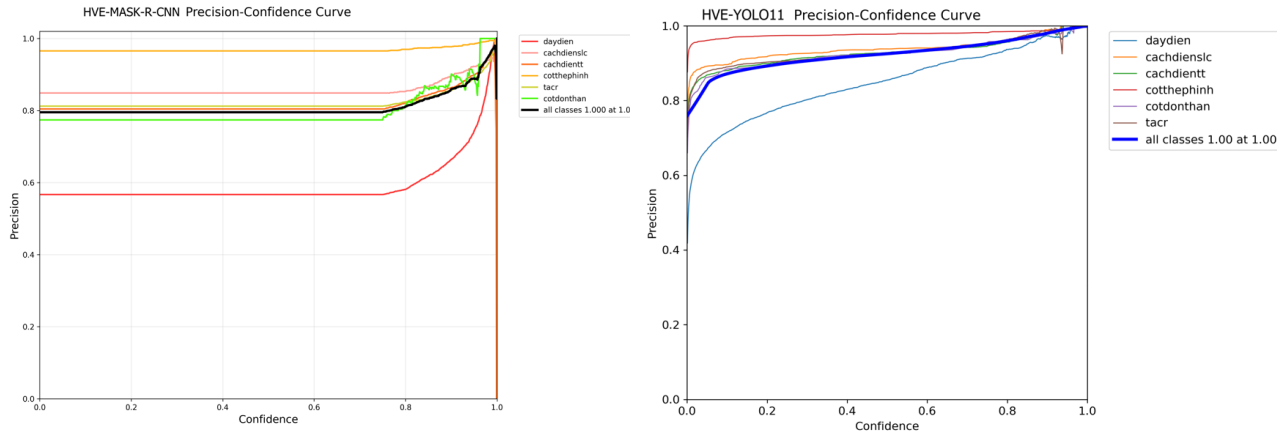


Fig. 12. Precision-Confidence curve analysis

Based on the analysis of the Recall-Confidence curves (Fig. 13), a detailed comparison of the comprehensive detection ability (recall) of the two models, HVE-MASK-R-CNN and HVE-YOLO11, can be made. The analysis shows that the HVE-YOLO11 model significantly outperforms HVE-MASK-R-CNN at finding all target objects. HVE-YOLO11 achieves nearly perfect recall. The overall curve (dark blue) starts with a recall of 0.99 at a confidence threshold of 0.0. This means that, with the default settings, the model can find 99% of the total objects in the data. Moreover, it maintains an extremely high recall (over 0.98) until the

confidence threshold reaches about 0.7. On the other hand, the HVE-MASK-R-CNN model shows a significantly lower overall recall. The overall curve (black) peaks at a recall of only 0.906. Although this is a good result, it indicates that even at the lowest confidence threshold, the model misses nearly 10% of the total objects.

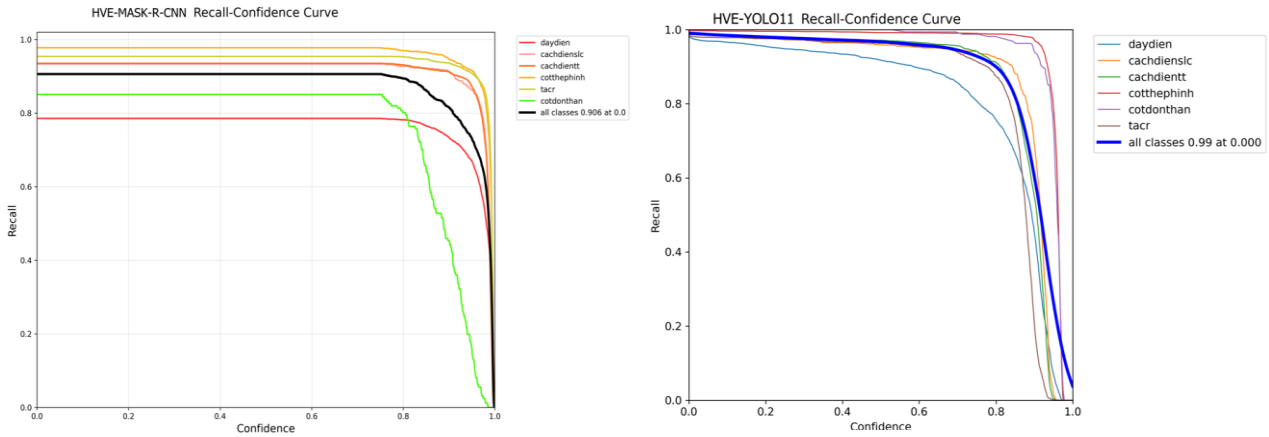


Fig. 13. Recall-Confidence curve analysis

The Precision-Recall (PR) curve (Fig. 14) is a comprehensive evaluation tool that illustrates the trade-off between Precision and Recall of a model. The area under this curve, known as Average Precision (AP), summarizes the model's performance for a specific class. The mean average precision (mAP) across all classes is the gold standard metric for comparing overall performance. Based on the analysis of the PR curves and mAP scores, the HVE-YOLO11 model outperforms the HVE-MASK-R-CNN model. The HVE-YOLO11 model achieves an mAP@0.5 score of 0.972, an exceptionally high value approaching perfection. The overall curve (dark blue) is nearly rectangular, hugging the top-right corner of the plot. This indicates that the model can maintain almost perfect precision (Precision ≈ 1.0) while detecting nearly all objects (Recall ≈ 0.9). In contrast, the HVE-MASK-R-CNN model achieves an mAP@0.5 of 0.855. While this is a good result, it is significantly lower than that of HVE-YOLO11. The overall curve (dashed black line) shows a sharp decline in precision as recall increases, especially above the 0.6 threshold.

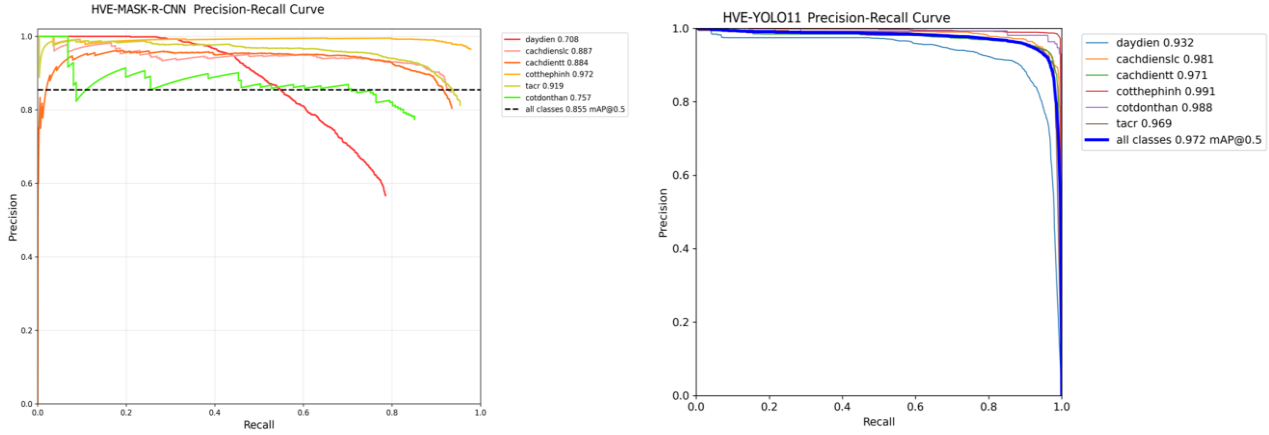
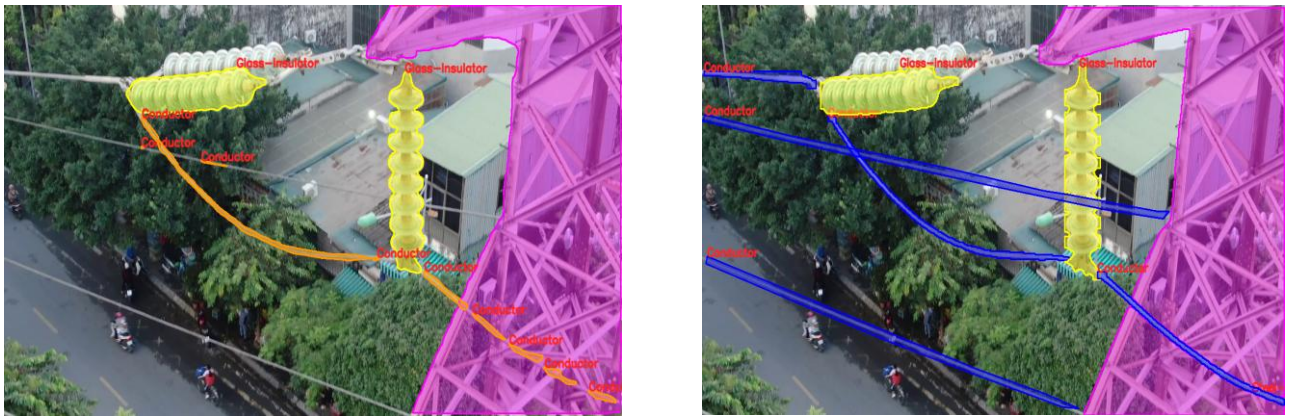


Fig. 14. Precision-Recall curve analysis

Qualitative visual inference analysis presented in Fig. 15 substantiates the statistical divergences observed in the Precision-Recall distributions. The actual results show that the HVE-MASK-R-CNN model missed some electrical wire objects (conductors), and, in addition, a single continuous wire was sometimes not detected as a whole but fragmented into many small, disconnected objects. This fragmentation occurs primarily because the RoIAlign pooling mechanism struggles to process extreme, frame-spanning aspect ratios, leading to localized spatial constraints. Conversely, the HVE-YOLO11 model missed fewer objects and detected the electrical cables far more accurately. It demonstrates immense robust spatial continuity when challenged with the electrical cable class, accurately enclosing extensive wire spans as single, continuous entities. This

morphological resilience is a direct functional outcome of the global context processing enabled by the C2PSA spatial attention module within the YOLO11 architecture. Both models, however, demonstrated high proficiency in producing accurate localized boundary fits for geometrically distinct and rigid objects, such as the glass insulators visible in the frame.



(a) Results detected by the HVE-MASK-R-CNN model **(b) Results detected by the HVE-YOLO11 model**

Fig. 15. Qualitative inference comparison

To evaluate processing speed under various operating conditions, the two models, HVE-YOLO11 and HVE-MASK-R-CNN, were tested using a series of GPU performance benchmarks (Fig. 16). Processing speed was measured in Frames Per Second (FPS), with higher FPS values indicating faster and more efficient image processing by the models. The empirical throughput superiority of HVE-YOLO11 over HVE-MASK-R-CNN, as observed during the GPU hardware benchmarking, is inextricably linked to the profound disparities in their underlying theoretical computational complexities.

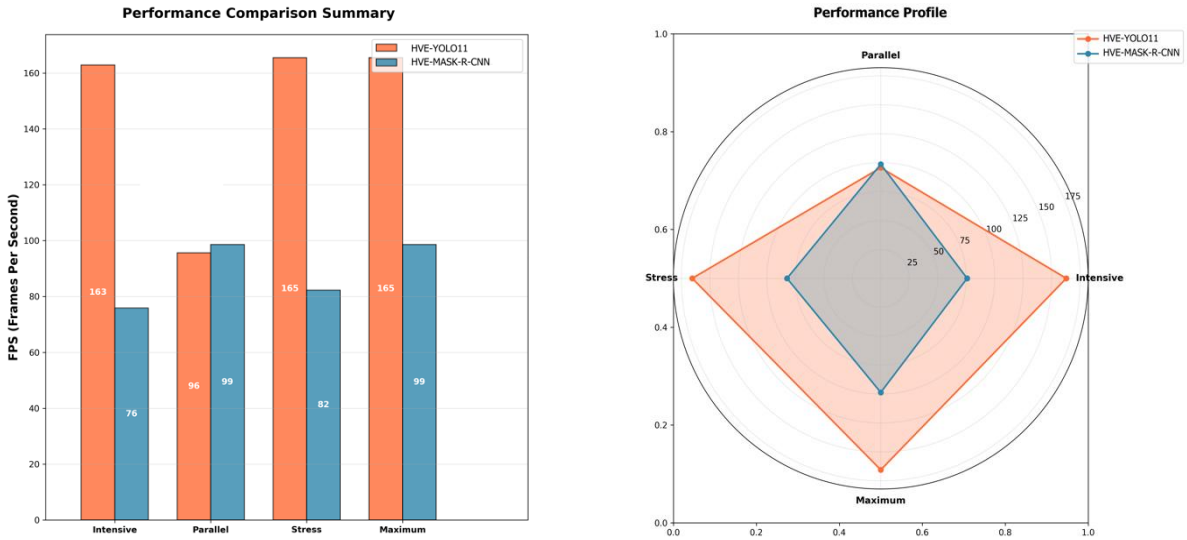


Fig. 16. The models' computational performance

As delineated in Table 2, the HVE-MASK-R-CNN architecture, encumbered by a deep ResNet-101-FPN backbone, necessitates approximately 63 million parameters. More critically, the two-stage nature of the network—requiring the sequential generation of region proposals followed by dense RoIAlign interpolations—results in a massive computational burden frequently exceeding 250 GFLOPs per inference at high resolutions. Conversely, the HVE-YOLO11 architecture achieves state-of-the-art mAP while relying on an exponentially smaller parameter footprint. The integration of the efficient C3k2 blocks optimizes gradient routing, significantly suppressing the baseline GFLOP requirement. Even when the mathematical complexity is subjected to a 2.25x scaling factor required to process the massive 1280×720 spatial resolution of the

UAV imagery, the computational overhead of YOLO11 remains a fraction of the two-stage model. This theoretical efficiency directly manifests in the empirical Intensive Test outcomes, where the minimized GFLOP load prevents GPU thermal throttling and memory bandwidth saturation, allowing HVE-YOLO11 to sustain 162.9 FPS compared to the bottlenecked 75.8 FPS of the Mask R-CNN model.

Tab. 2. Parametric Footprint and Theoretical Computational Complexity Matrix

Model Architecture	Network Backbone Strategy	Total Parameter Count	Theoretical GFLOPs (at 640^2)	Estimated GFLOPs (at 1280×720)
HVE-YOLO11 (Nano)	C3k2 + C2PSA	~2.6 Million	6.5	~14.6
HVE-YOLO11 (Medium)	C3k2 + C2PSA	~20.1 Million	68.0	~153.0
HVE-MASK-R-CNN	ResNet-101 + FPN	~63.0 Million	N/A	> 250.0 (Highly Variable)

The superior throughput of HVE-YOLO11 over HVE-MASK-R-CNN is fundamentally attributed to their different computational complexities. As shown in Tab. 2, HVE-MASK-R-CNN employs a heavy ResNet-101-FPN backbone with approximately 63 million parameters. Its two-stage pipeline, involving region proposal generation and RoIAlign operations, results in a computational cost exceeding 250 GFLOPs per inference at high resolutions. In contrast, HVE-YOLO11 achieves state-of-the-art mAP with a significantly smaller parameter footprint. The integration of efficient C3k2 blocks reduces computational overhead, enabling the model to remain lightweight even under the 2.25× scaling required for 1280×720 UAV imagery. This theoretical efficiency directly translates into hardware performance, preventing GPU thermal throttling and memory bandwidth saturation, thereby allowing HVE-YOLO11 to sustain 162.9 FPS compared to 75.8 FPS for HVE-MASK-R-CNN.

The models were evaluated under three scenarios: Intensive Test, Parallel Processing, and Stress Test. The Intensive Test measured raw inference throughput under heavy workloads, the Parallel Processing test evaluated multi-stream or batch processing capability, and the Stress Test assessed stability and sustained performance over 30 seconds.

HVE-YOLO11 demonstrated clear superiority in both Intensive and Stress Tests, achieving 162.9 FPS and 165.5 FPS, respectively, compared to 75.8 FPS and 82.3 FPS for HVE-MASK-R-CNN. These results confirm that YOLO11 is highly optimized for real-time and high-throughput applications.

In the Parallel Processing test, HVE-MASK-R-CNN slightly outperformed HVE-YOLO11 (98.6 FPS vs. 95.6 FPS). This behavior is attributed to the highly asynchronous computational graph of the two-stage detector, which allows the PyTorch/CUDA backend to efficiently vectorize fully connected classification operations across large batches of dynamically pooled RoIs on Tesla K80 GPUs. Consequently, the computational overhead of Mask R-CNN becomes highly parallelizable under large-batch execution, partially narrowing the throughput gap with the single-stage YOLO11 architecture.

Nevertheless, this marginal advantage does not offset the substantial overall speed superiority of HVE-YOLO11. Its peak throughput of 165.5 FPS is nearly 70% higher than the maximum 98.6 FPS achieved by HVE-MASK-R-CNN, highlighting a significantly higher performance ceiling. Overall, HVE-YOLO11 consistently outperforms HVE-MASK-R-CNN in high-intensity and continuous-processing scenarios, making it better suited for real-time monitoring, live video analysis, and large-scale UAV image processing.

5. CONCLUSIONS

This study conducted a comprehensive, quantitative comparison of two advanced object recognition architectures, HVE-YOLO11 (single-stage) and HVE-MASK-R-CNN (two-stage), for the specialized task of identifying six types of equipment on high-voltage power grids from UAV-captured images. The experimental results demonstrate that the HVE-YOLO11 model significantly outperforms the HVE-MASK-R-CNN model across both core metrics: accuracy and processing speed.

In terms of accuracy, HVE-YOLO11 achieved an mAP@0.5 score of 0.972, considerably higher than the 0.855 score of HVE-MASK-R-CNN. Further analysis through the F1-Confidence, Precision-Confidence, and Recall-Confidence curves reveals that HVE-YOLO11 not only achieves higher overall performance but also

demonstrates consistent stability and effectiveness across all object classes. Conversely, HVE-MASK-R-CNN exhibited a clear weakness in identifying classes with complex characteristics, such as 'electric cable' (elec-cable).

Regarding processing speed, GPU benchmark tests confirmed the overwhelming superiority of HVE-YOLO11, with inference speeds in intensive and stress tests more than double those of HVE-MASK-R-CNN (approximately 165 FPS vs. 82 FPS). This speed is a critical factor for meeting the requirements of real-time monitoring and analysis applications.

A key finding of this research is that, in this application, the single-stage HVE-YOLO11 model has broken the traditional trade-off between speed and accuracy, as it is not only faster but also significantly more accurate than the two-stage model.

In summary, with its combination of superior accuracy, stable performance across object classes, and real-time processing speed, HVE-YOLO11 is identified as the optimal model for deployment in automated inspection and monitoring systems for high-voltage power grids. Future research could focus on optimizing this model for deployment on edge devices and expanding the dataset to handle more complex operational scenarios.

Acknowledgments

This work was supported under the Industry 4.0 Research, Development, and Technology Application Support Program (code: KC-4.0.31/19-25).

Conflicts of Interest

The authors declare no conflict of interest.

REFERENCES

Chen, W., Li, Y., & Li, C. (2020). A visual detection method for foreign objects in power lines based on Mask R-CNN. *International Journal of Ambient Computing and Intelligence, 11*(1), 34–47. https://doi.org/10.4018/IJACI.2020010103

Chen, X., Xu, X., Xue, J., Zhang, W., & Wang, Q. (2024). A scene knowledge integrating network for transmission line multi-fitting detection. *Sensors, 24*(24), Article 8207. https://doi.org/10.3390/s24248207

Faisal, M. A. A., Mecheter, I., Qiblawey, Y., Hernandez Fernandez, J., Chowdhury, M. E. H., & Kiranyaz, S. (2025). Deep learning in automated power line inspection: A review. *Applied Energy, 385*, Article 125507. https://doi.org/10.1016/j.apenergy.2025.125507

Ji, Y., Zhang, D., He, Y., Zhao, J., Dong, X., & Zhang, T. (2025). Improved YOLO11 algorithm for insulator defect detection in power distribution lines. *Electronics, 14*(6), Article 1201. https://doi.org/10.3390/electronics14061201

Rong, S., He, L., Atici, S. F., & Cetin, A. E. (2025). Advanced YOLO-based real-time power line detection for vegetation management. *IEEE Transactions on Power Delivery, 40*(4), 2142–2153. https://doi.org/10.48550/arXiv.2503.00044

Santos, T., Tiago, C., André, D., António, P. M., & José, A. (2024). UAV visual and thermographic power line detection using deep learning. *Sensors, 24*(17), Article 5678. https://doi.org/10.3390/s24175678

Shao, Y., Zhang, R., Lv, C., Luo, Z., & Che, M. (2024). TL-YOLO: Foreign-object detection on power transmission line based on improved YOLOv8. *Electronics, 13*(8), Article 1543. https://doi.org/10.3390/electronics13081543

Tang, K., Peng, Y., Wu, X., & Qi, L. (2025). Improvement of Mask R-CNN algorithm for ore segmentation. *Electronics, 14*(10), Article 2025. https://doi.org/10.3390/electronics14102025

Vemula, S., & Frye, M. (2020). Mask R-CNN powerline detector: A deep learning approach with applications to a UAV. In *Proceedings of the 2020 IEEE/AIAA 39th Digital Avionics Systems Conference (DASC)* (pp. 1–6). IEEE. https://doi.org/10.1109/DASC50938.2020.9256456

Wang, C., Chen, Y., Wang, Z., Li, B., Tang, H., Jiang, D., & Sui, X. (2025). Line-YOLO: An efficient detection algorithm for power line angle. *Sensors, 25*(3), Article 876. https://doi.org/10.3390/s25030876

Wang, Z., Yuan, G., Zhou, H., Ma, Y., & Ma, Y. (2023). Foreign-object detection in high-voltage transmission line based on improved YOLOv8m. *Applied Sciences, 13*(23), Article 12775. https://doi.org/10.3390/app132312775

Wang, Z., Yuan, G., Zhou, H., Ma, Y., Ma, Y., & Chen, D. (2025). Improved YOLOv7 model for insulator defect detection. *arXiv*. https://doi.org/10.48550/arXiv.2502.07179

Wu, X., Sahoo, D., & Hoi, S. C. H. (2020). Recent advances in deep learning for object detection. *Neurocomputing, 396*, 39–64. https://doi.org/10.1016/j.neucom.2020.01.085

Xu, J., Zhang, S., Liu, Y., Sun, W., & Zhang, K. (2025). MRB-YOLOv8: An algorithm for insulator defect detection. *Electronics, 14*(5), Article 830. https://doi.org/10.3390/electronics14050830

Zhou, M., Wang, J., & Liu, B. (2022). ARG-Mask RCNN: An infrared insulator fault-detection network based on improved Mask RCNN. *Sensors, 22*(13), Article 4720. https://doi.org/10.3390/s22134720