Jornadas de Automática

Oriented object detection: edge models vs. cloud architectures

Eri Pérez Corral[1,*], Enrique Alegre Gutiérrez[1], Christopher Gaul[1], Milad Mirjalili[1], Waqar Tanveer[1]

[a]*Group for Vision and Intelligent Systems, I4 Institute, Universidad de León, 24071, León, Spain.*

To cite this article: Pérez, E., Alegre, E., Gaul, C., Mirjalili, M., Tanveer, W. 2026. Oriented object detection: edge models vs. cloud architectures. Jornadas de Automática, 47. https://doi.org/10.17979/ja-cea.2026.47.13834

Resumen

La detección de objetos en imágenes aéreas capturadas por drones requiere modelos capaces de localizar objetos con orientaciones arbitrarias y, al mismo tiempo, funcionar bajo las limitaciones de cómputo propias del procesamiento a bordo.

Este trabajo evalúa once modelos de detección de objetos orientados de las familias YOLO-OBB, RTMDet, RoI Transformer y Strip R-CNN.

La comparación considera tanto la precisión de detección como métricas relevantes para su uso en escenarios reales, como velocidad de procesamiento, tamaño del modelo y coste computacional. La evaluación se realiza sobre *parches* solapados del dataset DOTA v1.0. Los modelos se comparan mediante mAP y F1-Score ponderado usando umbrales de confianza seleccionados en el subconjunto de calibración. Los resultados muestran que los modelos de mayor capacidad, como YOLOv11x-OBB y RTMDet-l, son los candidatos más adecuados para inferencia en la nube, mientras que las variantes YOLO nano y RTMDet-tiny ofrecen distintos compromisos para el procesamiento en el borde. A partir de estos resultados, se plantea una arquitectura híbrida edge–cloud para detección orientada desde UAVs.

Palabras clave: Aprendizaje automático, Procesamiento de imágenes, Robots aéreos, Percepción y sensado, Sistemas embebidos de control, UAVs

Abstract

Aerial object detection from UAV imagery requires models capable of localizing objects with arbitrary orientations while operating under the computational constraints of onboard processing. This work evaluates eleven oriented object detection models from the YOLO-OBB, RTMDet, RoI Transformer and Strip R-CNN families.

The comparison considers both detection accuracy and deployment-oriented metrics, including latency, throughput, model size and computational complexity. The evaluation uses overlapping image patches from the DOTA v1.0 dataset. Models are compared in terms of mAP and weighted F1 using confidence thresholds selected on the calibration subset. The results show that high-capacity models such as YOLOv11x-OBB and RTMDet-l are the most suitable candidates for cloud-side inference, whereas the YOLO nano variants and RTMDet-tiny offer different trade-offs for edge-side processing. Based on these findings, a hybrid edge–cloud architecture is proposed, where lightweight onboard detectors process the UAV video stream in real time and more accurate cloud-side models are reserved for ambiguous, low-confidence or high-priority detections.

Keywords: Machine learning, Image processing, Flying robots, Perception and sensing, Embedded computer control systems and applications, UAVs

1. Introduction

Object detection in aerial imagery has become a key task in remote sensing and UAV-based monitoring applications. It enables the automatic localization and classification of relevant objects such as vehicles, ships, aircraft, bridges, and infrastructures (Tang et al., 2023). These capabilities are useful

*Corresponding author: [eperec@unileon.es]

in scenarios such as traffic monitoring, emergency response, environmental supervision, precision agriculture, and surveillance. However, aerial object detection presents specific challenges that differ from conventional object detection in natural images, including small object size, dense object distributions, complex backgrounds, scale variability, and arbitrary object orientations (Ding et al., 2021).

Most general-purpose object detectors rely on horizontal bounding boxes (HBBs), which are not well suited to aerial imagery when objects appear with arbitrary orientations or dense spatial layouts (Xu et al., 2020). Oriented bounding boxes (OBBs) provide a more appropriate representation by modelling object rotation and better fitting the actual spatial extent of aerial targets.

In addition to detection accuracy, UAV-based applications impose strong deployment constraints. Onboard systems are limited by computational resources, memory, energy consumption, and real-time inference requirements. Lightweight models can provide fast inference on edge devices, but may suffer from lower localization accuracy or reduced robustness in complex scenes (Yuan et al., 2024). Conversely, high-capacity models usually achieve better detection performance, but are more suitable for cloud-side execution due to their computational cost. This trade-off motivates the study of hybrid edge–cloud architectures, where a lightweight detector performs fast onboard inference and a more accurate cloud model is used for selected frames, uncertain detections or refinement.

This work evaluates oriented object detection models from two complementary frameworks: Ultralytics YOLO-OBB and MMRotate. The selected models cover both lightweight edge-oriented detectors and high-capacity cloud-oriented architectures. The comparison considers not only standard detection metrics, but also deployment-related indicators including latency, throughput, model size, number of parameters, and computational complexity. In addition, an F1 analysis is performed to study the practical operating point of each detector under different confidence thresholds.

The rest of the paper is structured as follows. Section 2 provides an overview of related research on aerial object detection, oriented bounding boxes, and deployment-oriented detection. Section 3 details the experimental setup, including the dataset, evaluated models, metrics, and implementation settings. Section 4 reports and discusses the obtained results. Section 5 presents the proposed hybrid edge–cloud architecture. Finally, Section 6 concludes the paper and highlights potential avenues for future work.

2. Related Work and State of the Art

2.1. Aerial object detection and datasets

Aerial object detection has become increasingly relevant due to the growth of UAV-based image acquisition, with applications to urban planning, precision agriculture, emergency response, and surveillance (Zhu et al., 2018; Sadgrove et al., 2018; Reilly et al., 2010), among others. Compared to natural-image detection, aerial scenes present additional challenges including bird's-eye viewpoints, arbitrary object orientations, large scale variations, dense spatial distributions, and small or elongated objects (Xia et al., 2018; Ding et al., 2019, 2021).

Most general-purpose detectors rely on horizontal bounding boxes, which are not very appropriate for aerial imagery when objects appear with arbitrary orientations or dense layouts (Ding et al., 2019; Xu et al., 2020). Oriented bounding boxes (OBBs) provide a better representation by modelling object rotation and fitting the actual spatial extent of aerial targets (Xia et al., 2018; Zhu et al., 2015). The DOTA dataset (Xia et al., 2018) addressed the lack of large-scale annotated benchmarks for this task, providing multi-source high-resolution imagery with oriented bounding box annotations across 15 object categories, and has become the standard benchmark for oriented object detection in remote sensing.

2.2. Deep Learning Architectures for OBB Detection

Modern object detection has evolved from two-stage region-based detectors to efficient one-stage architectures. Faster R-CNN introduced region proposal networks within a unified detection framework (Ren et al., 2015), while YOLO (Redmon et al., 2016) reformulated detection as a single-stage regression problem suitable for real-time inference. RetinaNet (Lin et al., 2017) further improved dense one-stage detection by introducing focal loss to address foreground-background imbalance. These general-purpose architectures have strongly influenced aerial object detection, but the arbitrary orientation and dense distribution of aerial objects motivated specific adaptations for rotated object detection.

RoI Transformer (Ding et al., 2019) is one of the representative two-stage approaches for oriented object detection in aerial images. It learns to transform horizontal regions of interest into rotated regions of interest, reducing the mismatch between axis-aligned proposals and arbitrarily oriented objects. More recent remote sensing detectors have focused on improving feature extraction for aerial scenes. For instance, LSKNet adapts the receptive field using large selective kernels (Li et al., 2023), PKINet combines multi-scale convolutional kernels with contextual attention (Cai et al., 2024), and Strip R-CNN exploits strip convolutions to better model elongated high-aspect-ratio objects (Yuan et al., 2026).

In parallel, efficient one-stage detectors have also been extended to rotated object detection. RTMDet (Lyu et al., 2022) provides real-time detection variants within the OpenMMLab ecosystem, whereas YOLO-based OBB models preserve the deployment-oriented nature of YOLO architectures while adding support for oriented bounding boxes (Ultralytics, 2026). These models are particularly relevant for UAV-based scenarios, where detection accuracy must be balanced against latency, model size, and computational cost.

2.3. Edge AI and Deployment-Oriented Detection

Although high-capacity oriented detectors can achieve strong accuracy, their computational cost may limit their use in onboard UAV systems. UAV-based aerial surveillance requires processing high-resolution image streams under constraints of latency, memory, energy consumption, bandwidth, and embedded hardware capacity. Edge AI addresses this problem by deploying machine learning models close to the data source, reducing the need to continuously transmit raw data to remote servers (Gill et al., 2025).

From a deployment perspective, lightweight one-stage detectors are attractive because they usually provide lower latency and simpler inference pipelines than multi-stage architectures (Redmon et al., 2016; Lyu et al., 2022). Additional optimization techniques, such as reduced-precision inference, ONNX export and TensorRT acceleration, can further improve inference speed and memory usage on NVIDIA-based hardware (NVIDIA, 2026).

However, choosing between a highly accurate but computationally expensive model and a fast but less robust lightweight detector remains a fundamental compromise. This trade-off motivates collaborative edge–cloud strategies, where computationally cheap inference is performed onboard and more demanding analysis is delegated to cloud-side models when necessary (Yuan et al., 2024).

3. Experimental Setup

The experimental evaluation was designed to compare oriented object detection models under both accuracy-oriented and deployment-oriented criteria. Therefore, the models were compared not only in terms of mAP, but also considering latency, throughput, model size, number of parameters, and computational complexity.

3.1. Dataset and Preprocessing

The evaluation was conducted on the DOTA v1.0 validation set (Xia et al., 2018). The original dataset comprises 2,806 high-resolution images, with dimensions ranging from 800×800 to $20{,}000 \times 20{,}000$ pixels, distributed across 15 class-imbalanced categories (e.g., planes, ships, helicopters, and small and large vehicles).

The official validation split was divided into two disjoint subsets: a calibration subset and a held-out test subset. The split was performed at the original-image level, using the source-image identifier preserved in the patch filenames. Therefore, all patches derived from the same high-resolution image were assigned to the same subset, preventing overlap leakage between calibration and test data. Due to the large size of the original images, we processed them into fixed-size patches following the standard MMRotate preprocessing guidelines (Zhou et al., 2022). Specifically, the images were cropped into 1024×1024 pixel patches with a sliding window overlap of 200 pixels. For the validation split, this procedure generated a total of 5,297 image patches. Out of these, 3,066 patches contained positive annotations, while the remaining 2,231 were empty. The complete validation split, including images with empty annotation files, was intentionally preserved during the evaluation phase to rigorously test the models' robustness and appropriately penalize false positive predictions in background-only regions.

3.2. Evaluated Models

The evaluation was conducted across two distinct deep learning frameworks: Ultralytics (Ultralytics, 2026), a widely adopted environment renowned for its one-stage YOLO architectures with different model sizes, and MMRotate (Zhou et al., 2022), an open-source, PyTorch-based toolbox dedicated to rotated object detection, developed as part of the OpenMMLab ecosystem.

We selected a diverse set of models from both ecosystems, ranging from lightweight edge-oriented architectures to heavy, high-accuracy cloud models. Within the YOLO-OBB family, six variants were evaluated: YOLOv8n-OBB, YOLOv8x-OBB, YOLOv11n-OBB, YOLOv11x-OBB, YOLOv26n-OBB and YOLOv26x-OBB. These models represent three Ultralytics generations evaluated at two capacity levels. YOLOv8-OBB is included as an established deployment-oriented baseline within the Ultralytics ecosystem. YOLOv11-OBB represents a more recent generation aimed at improving the accuracy–efficiency balance, while YOLOv26-OBB is included as a newer Ultralytics candidate offering reduced computational cost compared to previous generations. For each generation, the nano variant is considered an edge candidate due to its reduced size and latency, whereas the extra-large variant is considered a cloud candidate due to its higher capacity.

The MMRotate benchmark included five representative rotated object detection models: RoI Transformer-R50, RoI Transformer-Swin-Tiny, Strip R-CNN-S, RTMDet-tiny and RTMDet-l. RoI Transformer-R50 was included as a classical two-stage oriented detection baseline, while RoI Transformer-Swin-Tiny was used to assess the effect of a transformer-based backbone. Strip R-CNN-S was selected as a high-capacity remote sensing architecture specifically designed for elongated and oriented objects. Finally, RTMDet-tiny and RTMDet-l were included to evaluate lightweight and high-capacity variants of a recent one-stage rotated detector family.

3.3. Evaluation metrics

The evaluation combines both accuracy-oriented and deployment-oriented metrics to analyze the models and their suitability for edge and cloud architectures.

Since the evaluated models belong to two different software ecosystems, the metrics were computed according to the corresponding validation pipeline of each framework. The results are reported separately for both ecosystems and interpreted mainly in terms of trends and deployment suitability rather than as a strict one-to-one framework ranking.

3.3.1. Detection Accuracy Metrics

Detection accuracy was evaluated using mean Average Precision (mAP) at different rotated Intersection over Union (IoU) thresholds, specifically mAP50, mAP75, and mAP50-95. In oriented object detection, IoU is computed between rotated bounding boxes rather than horizontal boxes, so localization quality is assessed according to the overlap between predicted and ground-truth oriented boxes.

In addition, precision, recall, and F1-score were computed from true positives, false positives and false negatives. Given the significant class imbalance inherent to the DOTA dataset, weighted F1-score was used as the primary fixed-threshold metric, which averages class-wise results proportionally to their ground-truth instance counts.

Fixed-threshold metrics depend strongly on the confidence threshold used to accept or discard detections. To analyze this sensitivity, a confidence-threshold sweep was performed (from 0.05 to 0.95) to identify the optimal weighted F1-score for each model and IoU threshold. Predictions were matched greedily to ground-truth objects in descending confidence order using rotated IoU, with unmatched predictions counted as false positives and unmatched ground-truth objects as false negatives.

3.3.2. Deployment-Oriented Metrics

In addition to detection accuracy, several deployment-oriented metrics were reported. Latency measures the average inference time per image, while throughput is reported in frames per second (FPS). Model size refers to the storage size of the trained checkpoint or exported weights. The number of parameters provides an estimate of the memory footprint of the neural network, and GFLOPs estimate the computational cost of a forward pass.

Latency and throughput were measured with batch size 1 on the same NVIDIA A100-PCIE-40GB GPU, in order to approximate a real-time UAV video stream scenario where frames are processed sequentially. For YOLO-OBB models, these values were obtained from the Ultralytics validation logs. For models evaluated with MMRotate, latency was measured through repeated inference, excluding disk I/O and including detector heads and post-processing. Since each framework uses its native inference pipeline, latency values are interpreted as deployment-oriented indicators rather than as a fully unified cross-framework runtime benchmark. Parameters and GFLOPs were obtained using the corresponding analysis utilities of each framework.

These metrics are relevant for the proposed edge–cloud scenario: edge-side models must prioritize low latency, reduced memory usage and low computational cost, whereas cloud-side models can prioritize higher detection accuracy at the expense of increased resource consumption.

3.4. Implementation details

All experiments were executed in a GPU-based environment using Apptainer containers, with latency and throughput measured on an NVIDIA A100-PCIE-40GB GPU. The Ultralytics YOLO-OBB models were evaluated using the native Ultralytics validation pipeline, while the MMRotate models were evaluated using two different OpenMMLab environments. RoI Transformer, RoI Transformer with Swin-Tiny backbone and Strip R-CNN were evaluated using the legacy MMRotate framework, whereas RTMDet-tiny and RTMDet-l were evaluated using MMRotate 1.x.

All models were evaluated using single-scale inference with an input resolution of 1024×1024 pixels.

4. Results and Discussion

Following the experimental setup outlined in Section 3, the results are presented separately for the Ultralytics YOLO-OBB and MMRotate benchmarks. The analysis focuses on the accuracy–efficiency trade-off of each model family and on their suitability for the edge and cloud components of the proposed hybrid architecture.

4.1. Ultralytics YOLO-OBB Candidate Models

Table 1 summarizes the accuracy, confidence-threshold calibrated F1-score, and deployment characteristics of the YOLO-OBB models. Among the cloud-oriented variants, YOLOv11x-OBB achieves the strongest overall AP performance and the best weighted F1-score at IoU 0.50, making it the most accurate YOLO candidate in the evaluated set. YOLOv26x-OBB, however, obtains the best weighted F1-score at IoU 0.75 and the lowest computational cost among the extra-large variants, suggesting a competitive behaviour when stricter localization and efficiency are prioritized.

For the edge-oriented variants, YOLOv11n-OBB provides the best AP-based performance and the highest weighted F1-score at IoU 0.50, making it the most accurate lightweight YOLO model. YOLOv8n-OBB achieves the lowest latency and highest throughput, while YOLOv26n-OBB has the lowest parameter count and GFLOPs. This shows that the best edge candidate depends on the deployment priority: YOLOv11n-OBB is preferable when accuracy is prioritized, YOLOv8n-OBB when latency is critical, and YOLOv26n-OBB when computational cost is the main constraint.

The confidence-threshold sweep also reveals that the YOLO-OBB models generally achieve their best F1-scores at relatively high confidence thresholds, especially at IoU 0.75. This suggests that low-confidence detections tend to introduce false positives and that a more conservative operating point improves the precision–recall balance. However, the optimal threshold should not be interpreted as a direct measure of model quality, since confidence scores are not necessarily calibrated in the same way across model generations.

Overall, YOLOv11x-OBB is selected as the strongest YOLO cloud candidate, while YOLOv11n-OBB provides the best accuracy-oriented edge configuration. YOLOv8n-OBB and YOLOv26n-OBB remain relevant alternatives when the deployment priority shifts from accuracy to latency or computational efficiency.

4.2. Architectures Evaluated in the MMRotate Framework

The architectures evaluated with MMRotate cover a broader range of detection paradigms, including two-stage oriented detectors, remote-sensing-specific models and one-stage real-time detectors. As shown in Table 2, RTMDet-l achieves the strongest overall detection performance across the MMRotate group, leading both AP-based metrics and threshold-calibrated F1-scores. This makes it the most suitable MMRotate candidate for cloud-side inference.

RTMDet-tiny also achieves highly competitive results despite its substantially reduced capacity. It outperforms both RoI Transformer variants and exceeds slightly Strip R-CNN-S while being considerably more efficient. In terms of deployment, RTMDet-tiny is the most favourable MMRotate model, with the lowest latency, highest throughput, smallest checkpoint size, fewest parameters, and lowest computational cost. Therefore, it is selected as the most suitable MMRotate candidate for edge-side inference.

Strip R-CNN-S reaches competitive F1 values after confidence-threshold calibration, especially at IoU 0.50. However, its performance decreases under stricter localization

Table 1: Accuracy, confidence-threshold calibrated F1-score, and deployment characteristics of YOLO-OBB candidate models on the DOTA v1.0 validation split. AP metrics are obtained from the native Ultralytics validation pipeline. For each IoU threshold, the first column reports the best weighted F1-score obtained across the confidence-threshold sweep, and the second column (Thr.) indicates the corresponding optimal confidence threshold.

Model	**Role**	**mAP$_{50}$**	**mAP$_{75}$**	**mAP$_{50–95}$**	**F1$_w$@0.50**	**Thr.**	**F1$_w$@0.75**	**Thr.**	**Latency (ms)**	**FPS**	**Size (MB)**	**Params (M)**	**GFLOPs**
YOLOv8x-OBB	Cloud	83.99	77.28	68.62	76.25	0.65	62.59	0.70	21.19	47.20	139.56	69.46	673.86
YOLOv11x-OBB	Cloud	**85.19**	**78.84**	**69.74**	**77.11**	0.60	63.33	0.70	21.29	46.96	**118.44**	58.75	519.15
YOLOv26x-OBB	Cloud	83.28	76.97	68.65	76.12	0.65	**63.95**	0.75	**20.16**	**49.60**	126.94	**57.56**	**516.49**
YOLOv8n-OBB	Edge	80.31	73.03	63.54	73.19	0.55	57.54	0.65	**11.11**	**90.03**	6.57	3.08	21.36
YOLOv11n-OBB	Edge	**81.15**	**73.63**	**64.17**	**74.04**	0.50	56.98	0.65	12.45	80.31	**5.80**	2.66	16.83
YOLOv26n-OBB	Edge	79.68	73.08	63.85	72.24	0.50	**57.87**	0.65	11.93	83.84	5.91	**2.45**	**13.95**

Table 2: Accuracy, calibrated fixed-threshold F1-score, and deployment characteristics of candidates evaluated with MMRotate on the held-out DOTA v1.0 test subset. AP metrics are obtained from the native MMRotate evaluation pipeline. For each IoU threshold, the first column reports the weighted F1-score computed on the held-out test subset, and the second column (Thr.) indicates the confidence threshold previously selected on the calibration subset.

Model	**Role**	**mAP$_{50}$**	**mAP$_{75}$**	**mAP$_{50–95}$**	**F1$_w$@0.50**	**Thr.**	**F1$_w$@0.75**	**Thr.**	**Latency (ms)**	**FPS**	**Size (MB)**	**Params (M)**	**GFLOPs**
RTMDet-l	Cloud	**94.42**	**77.82**	**68.06**	**93.59**	0.40	**84.78**	0.50	25.58	39.09	444.51	52.27	205.00
Strip R-CNN-S	Cloud	88.19	69.70	60.43	90.13	0.75	74.60	0.80	46.18	21.65	430.10	45.07	218.62
RTMDet-tiny	Edge	89.39	65.73	58.80	89.61	0.40	76.85	0.50	**20.71**	**48.29**	**48.46**	**4.88**	**20.58**
RoI Transf.-Swin-T	Cloud	86.64	62.80	57.92	88.06	0.60	67.80	0.80	55.33	18.07	235.31	58.75	229.53
RoI Transf.-R50	Cloud	83.98	59.23	54.81	86.34	0.55	65.90	0.80	37.18	26.89	221.70	55.35	225.29

requirements and its deployment cost remains high. This suggests that Strip R-CNN-S benefits from a conservative operating point, but is less attractive than RTMDet-l for cloud inference and less suitable than RTMDet-tiny for edge deployment.

The RoI Transformer variants obtain the lowest overall performance among the MMRotate models. RoI Transformer-Swin-Tiny improves over RoI Transformer-R50, confirming that the transformer-based backbone provides a meaningful accuracy gain over the ResNet-50 version. However, both variants remain below RTMDet-tiny in AP-based metrics and calibrated F1. From a deployment perspective, their higher latency and weaker accuracy–efficiency trade-off make them less suitable for the proposed edge–cloud scenario.

The confidence-threshold sweep also reveals different operating behaviours across models. RTMDet-l and RTMDet-tiny reach their best F1-scores at moderate thresholds, while Strip R-CNN-S and the RoI Transformer variants require more conservative thresholds. This suggests that lower-confidence detections from these models are less reliable and must be filtered more aggressively to maximize F1.

Overall, RTMDet is the strongest MMRotate family in the evaluated set: RTMDet-l is selected as the cloud-side candidate, while RTMDet-tiny provides the best MMRotate edge-side configuration.

4.3. Cross-framework Discussion

Although both frameworks use different validation pipelines, the results show consistent deployment-oriented trends. Lightweight models provide lower latency, smaller checkpoints, and reduced computational cost, whereas high-capacity models generally achieve stronger detection performance. Therefore, Ultralytics and MMRotate should not be interpreted as a strict one-to-one ranking, but rather as complementary references for analyzing the accuracy–efficiency trade-off in oriented aerial object detection.

From an edge-deployment perspective, the reported measurements suggest that the YOLO nano variants are the most compact and fastest candidates, while RTMDet-tiny offers a stronger accuracy profile within MMRotate at a higher deployment cost. However, since latency was obtained through framework-specific pipelines, these values should be interpreted as deployment-oriented indicators rather than as a fully unified runtime benchmark.

These findings motivate the hybrid edge–cloud architecture described in Section 5, where lightweight models perform onboard inference and high-capacity models are reserved for selected frames or uncertain detections.

5. Hybrid Edge–Cloud Architecture

Based on the accuracy–efficiency trade-offs analyzed in Section 4, this work proposes a hybrid edge–cloud architecture for UAV-based oriented object detection. This approach is motivated by the observation that no single model simultaneously optimizes detection accuracy, latency, and computational cost. A purely edge-based solution enables real-time onboard processing but is limited by the computational resources of the UAV, whereas a purely cloud-based solution can exploit more accurate models but introduces bandwidth and latency dependencies.

The proposed deployment workflow is illustrated in Figure 1. It assigns different roles to lightweight and high-capacity detectors. On the UAV, an edge-oriented model, such as YOLOv11n-OBB, YOLOv8n-OBB, YOLOv26n-OBB, or RTMDet-tiny, processes the video stream in real time and generates preliminary detections. According to the deployment priority, the edge model can be selected to favour accuracy, latency, or computational cost. Frames or cropped regions corresponding to low-confidence detections can then be transmitted to the cloud, where a higher-capacity model such as YOLOv11x-OBB or RTMDet-l performs more accurate refinement.

This division of roles reduces the need to transmit the complete video stream while preserving access to more accurate cloud-side inference when needed. In this way, the architecture combines fast onboard detection with selective

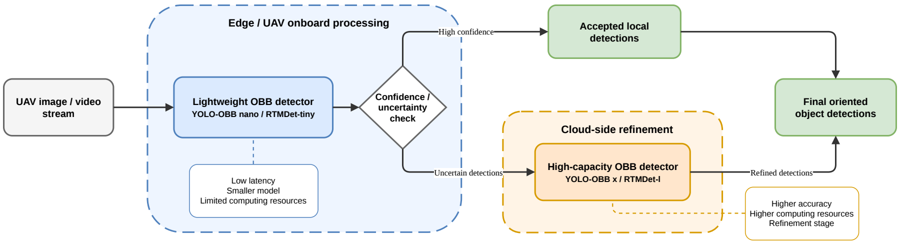

Figure 1: Proposed hybrid edge–cloud pipeline for oriented object detection in UAV imagery. A lightweight detector performs fast onboard inference, while uncertain detections or selected frames are forwarded to a cloud-side detector for refinement.

high-precision processing, making it suitable for UAV scenarios where both real-time response and detection quality are required.

6. Conclusions and Future Work

This work evaluated oriented object detection models from the Ultralytics YOLO-OBB and MMRotate ecosystems under both accuracy-oriented and deployment-oriented criteria. The results show that high-capacity models provide the strongest detection performance, with YOLOv11x-OBB and RTMDet-l emerging as the most suitable cloud-side candidates within their respective frameworks. In contrast, lightweight models offer more favourable deployment characteristics: YOLOv11n-OBB provides the best accuracy-oriented edge configuration among the YOLO variants, while YOLOv8n-OBB and YOLOv26n-OBB are preferable when latency or computational cost is prioritized. Within MMRotate, RTMDet-tiny provides the best edge-oriented accuracy–efficiency compromise. The cross-framework comparison should be interpreted with caution, as mAP values are obtained through different evaluation pipelines.

Future work will validate the selected edge candidates on embedded hardware, such as NVIDIA Jetson or Raspberry Pi platforms, since the reported latency values were measured on a server-grade NVIDIA A100 GPU. In addition, TensorRT, mixed-precision and INT8 optimization will be studied to evaluate whether high-capacity models such as RTMDet-l can become competitive edge candidates without compromising OBB localization accuracy.

Acknowledgements

The authors would like to thank the GVIS group of the University of León for their support and for providing the computational resources used in this work.

References

Cai, X., Lai, Q., Wang, Y., Wang, W., Sun, Z., Yao, Y., 2024. Poly kernel inception network for remote sensing detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 27706–27716.

Ding, J., Xue, N., Long, Y., Xia, G.-S., Lu, Q., 2019. Learning RoI transformer for oriented object detection in aerial images. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2849–2858.

Ding, J., Xue, N., Xia, G.-S., Bai, X., Yang, W., Yang, M. Y., Belongie, S., Luo, J., Datcu, M., Pelillo, M., et al., 2021. Object detection in aerial images: A large-scale benchmark and challenges. IEEE transactions on pattern analysis and machine intelligence 44 (11), 7778–7796.

Gill, S. S., Golec, M., Hu, J., Xu, M., Du, J., Wu, H., Walia, G. K., Murugesan, S. S., Ali, B., Kumar, M., et al., 2025. Edge AI: A taxonomy, systematic review and future directions. Cluster Computing 28 (1), 18.

Li, Y., Hou, Q., Zheng, Z., Cheng, M.-M., Yang, J., Li, X., 2023. Large selective kernel network for remote sensing object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 16794–16805.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988.

Lyu, C., Zhang, W., Huang, H., Zhou, Y., Wang, Y., Liu, Y., Zhang, S., Chen, K., 2022. RTMDet: An empirical study of designing real-time object detectors. arXiv preprint arXiv:2212.07784.

NVIDIA, 2026. NVIDIA TensorRT Documentation. `https://docs.nvidia.com/deeplearning/tensorrt/latest/index.html`, accessed: 2026-05-30.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788.

Reilly, V., Idrees, H., Shah, M., 2010. Detection and tracking of large number of targets in wide area surveillance. In: European conference on computer vision. Springer, pp. 186–199.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems 28.

Sadgrove, E. J., Falzon, G., Miron, D., Lamb, D. W., 2018. Real-time object detection in agricultural/remote environments using the multiple-expert colour feature extreme learning machine (MEC-ELM). Computers in Industry 98, 183–191.

Tang, G., Ni, J., Zhao, Y., Gu, Y., Cao, W., 2023. A survey of object detection for uavs based on deep learning. Remote Sensing 16 (1), 149.

Ultralytics, 2026. Ultralytics documentation. `https://docs.ultralytics.com/`, accessed: 2026-05-30.

Xia, G.-S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L., 2018. DOTA: A large-scale dataset for object detection in aerial images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3974–3983.

Xu, Y., Fu, M., Wang, Q., Wang, Y., Chen, K., Xia, G.-S., Bai, X., 2020. Gliding vertex on the horizontal bounding box for multi-oriented object detection. IEEE transactions on pattern analysis and machine intelligence 43 (4), 1452–1459.

Yuan, X., Zheng, Z., Li, Y., Liu, X., Liu, L., Li, X., Hou, Q., Cheng, M.-M., 2026. Strip R-CNN: Large strip convolution for remote sensing object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 40. pp. 12259–12267.

Yuan, Y., Gao, S., Zhang, Z., Wang, W., Xu, Z., Liu, Z., 2024. Edge-cloud collaborative UAV object detection: Edge-embedded lightweight algorithm design and task offloading using fuzzy neural network. IEEE Transactions on Cloud Computing 12 (1), 306–318.

Zhou, Y., Yang, X., Zhang, G., Wang, J., Liu, Y., Hou, L., Jiang, X., Liu, X., Yan, J., Lyu, C., et al., 2022. MMRotate: A rotated object detection benchmark using pytorch. In: Proceedings of the 30th ACM international conference on multimedia. pp. 7331–7334.

Zhu, H., Chen, X., Dai, W., Fu, K., Ye, Q., Jiao, J., 2015. Orientation robust object detection in aerial images using deep convolutional neural network. In: 2015 IEEE international conference on image processing (ICIP). IEEE, pp. 3735–3739.

Zhu, P., Wen, L., Bian, X., Ling, H., Hu, Q., 2018. Vision meets drones: A challenge. arXiv preprint arXiv:1804.07437.