

## RESEARCH ARTICLE



# Evaluating RF-DETR and YOLOv26 for Rip Current Detection and Segmentation

Van Lam Ho<sup>1,\*</sup>, Van Khang Le<sup>2</sup>, Xuan Vinh Le<sup>3</sup>, Trong Doi Nguyen<sup>4</sup> and Trang-Thi Ho<sup>5</sup>

<sup>1</sup>Department of Information Technology, Quy Nhon University, Vietnam

<sup>2</sup>Department of Mathematics and Statistics, Quy Nhon University, Vietnam

<sup>3</sup>Office of Academic Affairs, Quy Nhon University, Vietnam

<sup>4</sup>Faculty of Natural Sciences, Quy Nhon University, Vietnam

<sup>5</sup>Department of Computer Science and Information Engineering, TamKang University, Taiwan, Province of China

**Abstract:** This application study evaluates two existing instance-segmentation frameworks, RF-DETR (Region Focused Detection Transformer) and YOLOv26, for rip current detection in complex marine imagery. Both models were fine-tuned from pretrained weights on the same 18,389 training images. YOLOv26 used the 4349 labeled images as its validation/evaluation split, whereas the RF-DETR training workflow divided these images into 2184 validation and 2165 internal-test images; the two RF-DETR subsets were combined for the final evaluation so that both models were assessed on the same 4349 labeled images. No architectural modification to RF-DETR or YOLOv26 is claimed. The study instead examines their observed segmentation accuracy, training behavior, and frame-wise inference speed on RipVIS and two unmanned aerial vehicle (UAV) videos. Under the configurations used, RF-DETR produced higher mask mean average precision (mAP) values, whereas YOLOv26 processed the two videos faster. Because the training schedules differ and the study does not include repeated trials, component ablations, embedded-platform tests, or broad environmental coverage, the results should be interpreted as an exploratory application comparison rather than evidence of a new algorithm or deployment readiness.

**Keywords:** video instance segmentation, rip current detection, detection transformers, YOLO-based segmentation, marine scene understanding

## 1. Introduction

Video instance segmentation (VIS) is a fundamental computer-vision task that aims to detect, segment, and track object instances across video sequences. Compared with image segmentation, VIS must also address temporal dynamics, object motion, occlusion, illumination changes, and background complexity. These challenges are particularly relevant to environmental monitoring and coastal surveillance [1, 2]. Rip current detection is difficult because rip currents often appear as diffuse, irregular regions whose visual characteristics vary with wave motion, lighting, and sediment distribution. Foam and wave patterns may exhibit similar visual features, increasing the risk of false positives [3, 4].

Recent advances in deep learning have improved segmentation performance. Convolutional neural network (CNN)-based approaches, including the YOLO family, are widely used because of their computational efficiency [3, 5]. However, their performance remains task- and configuration-dependent. Transformer-based

architectures such as the detection transformer (DETR) model long-range contextual relationships through self-attention and have achieved strong results in detection and segmentation tasks [6, 7].

RipVIS [1] is a benchmark for VIS in marine environments. It contains annotated coastal frames with dynamic water surfaces, high visual variability, and negative samples without rip currents. These characteristics make RipVIS suitable for examining segmentation performance under visually ambiguous conditions [8, 9].

In this study, we evaluate two existing segmentation frameworks: Region Focused Detection Transformer (RF-DETR) and YOLOv26. Both are fine-tuned from pretrained weights; no architectural changes or new mathematical formulation are introduced. The contribution is therefore an application-oriented evaluation on rip current imagery, including dataset preparation, a comparison of observed segmentation metrics, and frame-wise inference on two unmanned aerial vehicle (UAV) videos. RF-DETR and YOLOv26 are trained with different schedules, so the results are reported as configuration-specific observations rather than a controlled claim that one architecture is intrinsically superior.

\*Corresponding author: Van Lam Ho, Department of Information Technology, Quy Nhon University, Vietnam. Email: [hovanlam@qnu.edu.vn](mailto:hovanlam@qnu.edu.vn)

## 2. Literature Review

Early VIS approaches primarily relied on CNN-based architectures. Emil Brissman et al. [10] introduced a foundational VIS framework that jointly addressed object detection and temporal association. Although such approaches achieved promising results, CNN-based methods can have difficulty modeling long-range contextual relationships in highly dynamic environments.

More recently, transformer-based architectures have improved object detection and segmentation. Carion et al. [6] formulated object detection as direct set prediction using transformer encoders and decoders, eliminating anchor boxes and non-maximum suppression. Zhu et al. [7] proposed Deformable DETR, which focuses attention on a sparse set of spatial sampling locations. Conditional DETR [11], Anchor DETR [12], DN-DETR [13], and DAB-DETR [14] further improved convergence and query representation.

Transformer-based segmentation has also advanced image and video understanding. Wang et al. [4] proposed an end-to-end VIS transformer that integrates information across frames. Cheng et al. [15] introduced Mask2Former, which uses masked attention for semantic, instance, and panoptic segmentation. Foundation-model studies have also examined segmentation generalization across visual domains [16].

In parallel with transformer-based developments, the YOLO family has remained one of the most widely adopted frameworks for real-time object detection and segmentation due to its computational efficiency and end-to-end design. YOLOv7 [17] demonstrated significant improvements in real-time detection accuracy through trainable bag-of-freebies optimization strategies. YOLOv10 [18] and PP-YOLOE [19] further enhanced detection quality using improved label assignment and feature aggregation techniques. More recently, YOLOv26 has been introduced as an advanced evolution of the YOLO series with architectural refinements designed to improve both segmentation accuracy and inference efficiency [20–22]. Compared with earlier YOLO versions, YOLOv26 provides stronger multi-scale feature learning and improved convergence behavior, making it suitable for challenging segmentation tasks in dynamic environments.

Rip currents are recognized as one of the most hazardous coastal phenomena and represent a major cause of drowning incidents worldwide. However, detecting rip currents remains difficult because their appearance is often diffuse, irregular, and highly dependent on environmental conditions such as wave motion, sediment distribution, and illumination changes [23]. Traditional remote sensing and image-processing methods have been applied to rip current detection, but their robustness remains limited in highly dynamic coastal scenes [24].

To support research in this domain, Dumitriu and together with co-authors in [3] introduced a benchmark dataset and YOLO-based baseline for rip current segmentation, demonstrating the feasibility of deep learning approaches for coastal hazard detection. Recently, the RipVIS benchmark dataset was proposed to support VIS research in marine environments [1]. The dataset contains complex coastal scenes characterized by dynamic water surfaces, high visual ambiguity, and a large proportion of negative samples. These characteristics make RipVIS a challenging benchmark for evaluating segmentation robustness and temporal consistency in real-world conditions.

The trade-off between segmentation accuracy and inference efficiency remains insufficiently explored for coastal monitoring. This study therefore evaluates RF-DETR and YOLOv26 on RipVIS and analyzes segmentation metrics, convergence behavior, and video inference under the reported configurations.

The comparison characterizes the tested pipelines rather than establishing a complete state-of-the-art benchmark.

Related instance-segmentation research also includes Mask-Refined Region-based Convolutional Neural Network (R-CNN), which refines ROIAAlign and fuses multi-scale features to improve boundary detail [25]. More recent transformer-based methods, including Mask2Former [15], video transformer segmentation [4], and Segment Anything Model (SAM)-oriented studies [16], provide relevant methodological context. These methods were not re-trained here because equivalent RipVIS runs, preprocessing, and evaluation artifacts were not available; accordingly, they are discussed as related technology rather than used to claim a controlled state-of-the-art comparison.

## 3. Research Methodology

### 3.1. Research design

This study uses the publicly available RipVIS dataset. Both RF-DETR and YOLOv26 were fine-tuned from pretrained weights using the same 18,389 training images. For YOLOv26, the remaining 4349 labeled images form a validation/evaluation split, and all are used in the final evaluation. The RF-DETR training workflow requires three folders and therefore partitions those 4349 labeled images into 2184 validation images and 2165 internal-test images. For the final cross-model evaluation reported in this study, predictions and ground-truth labels from both RF-DETR subsets are combined, yielding the same 4349-image evaluation set used for YOLOv26. Thus, the models share the training-set size and final evaluation-set size, but they do not use identical validation splits during training.

The processed dataset uses COCO-format annotations for RF-DETR and YOLO-format annotations for YOLOv26. These formats support the requirements of the respective segmentation frameworks. Table 1 summarizes the resulting data organization.

Each architecture required a separate preprocessing pipeline. For RF-DETR, sampled images and annotations were reorganized into a COCO-style directory structure, checked for image-annotation consistency, and divided into training-time validation and internal-test splits. For YOLOv26, images and polygon labels were reorganized into the standard YOLO format, labels were validated, and empty label files were generated for negative images. Retaining negative samples helps the model learn to suppress false detections.

### 3.2. Participants

#### 3.2.1. RF-DETR

RF-DETR is a transformer-based segmentation model that extends the DETR framework [6, 7]. The model employs a convolutional backbone followed by a transformer encoder–decoder architecture.

To improve efficiency, RF-DETR integrates deformable attention mechanisms [20, 26], allowing the model to focus on relevant spatial regions. The use of object queries enables direct set prediction, eliminating the need for anchor boxes and non-maximum suppression.

**Mathematical Formulation of RF-DETR:** Given an input image  $I \in \mathbb{R}^{H \times W \times 3}$ , the RF-DETR framework aims to predict a set of object instances:

$$y = \{(c_i, b_i)\}_{i=1}^N \quad (1)$$

**Table 1**  
**Comparison of the RipVIS dataset before and after preprocessing for RF-DETR and YOLOv26**

| Aspect                                     | Original RipVIS                | YOLOv26 format                      | RF-DETR format                    |
|--------------------------------------------|--------------------------------|-------------------------------------|-----------------------------------|
| Annotation format                          | COCO + YOLO                    | YOLO polygon text files             | COCO JSON Annotations             |
| Training images                            | 18,389                         | 18,389                              | 18,389                            |
| Training-time validation/evaluation images | 4349 labeled evaluation images | 4349 labeled evaluation/test images | 2184 validation images            |
| Internal-test split                        | Not explicitly separated       | Not used                            | 2165 images split from validation |
| Negative samples                           | Implicitly present             | Empty label files generated         | Preserved in COCO annotations     |
| Additional training data                   | Available in repository        | Merged into training split          | Merged into training split        |
| Directory style                            | Mixed raw repository structure | Lightweight image/label folders     | Split-based COCO structure        |
| Main purpose                               | Raw source dataset             | Efficient YOLO training             | Transformer-ready training        |

where  $c_i$  denotes the class label and  $\mathbf{b}_i \in [0, 1]^4$  represents the normalized bounding box. Feature Extraction: A backbone network (e.g., ResNet or Swin Transformer) extracts feature maps:

$$F = B(I), \quad F \in R^{C \times H' \times W'} \quad (2)$$

Region Scoring and Selection: A lightweight scoring function is applied to estimate the objectness of each spatial location:

$$S = \sigma(F), \quad S \in R^{1 \times H' \times W'} \quad (3)$$

where  $\sigma(\cdot)$  denotes the sigmoid activation. We then select the top- $K$  regions:

$$R_K = \text{TopK}(S, K) \quad (4)$$

Each selected region corresponds to spatial coordinates:

$$R_K = \{(x_k, y_k)\}_{k=1}^K \quad (5)$$

Region-specific features are extracted using a sampling function  $\phi(\cdot)$ :

$$F_K = \phi(F, R_K), \quad F_K \in R^{K \times C} \quad (6)$$

The selected region features are encoded via a transformer encoder:

$$Z = \text{Encoder}(F_K) \quad (7)$$

Object Query Decoding: Let  $Q \in R^{N_q \times d}$  denote the learnable object queries. The decoder computes:

$$H = \text{Decoder}(Q, Z) \quad (8)$$

Final predictions are obtained as:

$$\hat{c}_i = \text{Softmax}(W_c h_i) \quad (9)$$

$$\hat{b}_i = \sigma(W_b h_i) \quad (10)$$

We define a bipartite matching  $\sigma$  between predictions and ground truth to set-based loss with Hungarian matching:

$$\sigma = \arg \arg \sum_{i=1}^N L_{\text{match}}(y_i, \widehat{y}_{\sigma(i)}) \quad (11)$$

The overall loss is:

$$L = \sum_{i=1}^N [L_{cls} + \lambda_1 L_{L1} + \lambda_2 L_{IoU}] \quad (12)$$

RF-DETR architecture used in this study: The equations above summarize the standard DETR/RF-DETR prediction, matching, and loss pipeline and are included only to describe the adopted model. They do not constitute a new formulation. We use the published RF-DETR segmentation architecture and pretrained weights without architectural modification and fine-tune the model on RipVIS. Its existing design uses region-focused/deformable feature aggregation and object queries to reduce the number of locations processed relative to dense global attention; these mechanisms are properties of RF-DETR, not contributions introduced in this study.

$$K \ll H' \times W' \quad (13)$$

The selected regions are converted into a compact token sequence  $\mathbf{F}_K$ , which serves as input to the transformer encoder. This reduces the quadratic complexity of self-attention:

$$O((H'W')^2) \rightarrow O(K^2) \quad (14)$$

Similar to DETR, RF-DETR employs a fixed number of learnable object queries. These queries attend to the encoded region features and produce a fixed-size set of predictions. In the published RF-DETR design, region-focused feature selection and deformable attention are intended to reduce the number of processed spatial locations and improve efficiency for high-resolution inputs. As illustrated in Figure 1, RF-DETR utilizes deformable attention to selectively aggregate features from relevant spatial locations, improving efficiency when processing high-resolution inputs. The use of object queries allows the model to predict a fixed number of object instances in a single forward pass.

In this paper, RF-DETR is used as an existing transformer-based segmentation model and is fine-tuned on RipVIS. The experiments evaluate its observed segmentation metrics and training behavior under the configuration in Table 2. Any association between its attention mechanisms and the measured results is treated as a possible explanation rather than a component-level causal conclusion, because no ablation study was conducted.

The RF-DETR-based rip current segmentation procedure is summarized below.

Figure 1  
The RF-DETR architecture

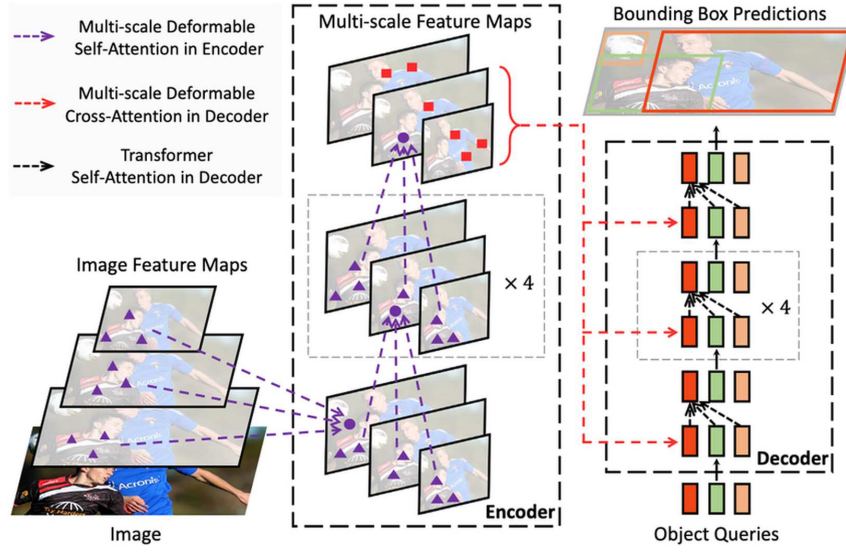


Table 2  
Main training configuration for RF-DETR

| Model          | Epochs | Batch | Resolution | Learning rate (LR) | Encoder learning rate |
|----------------|--------|-------|------------|--------------------|-----------------------|
| RF-DETR Medium | 10     | 8     | 432        | 0.0001             | 0.00015               |

**Input:** Video  $V$ , Trained RF-DETR model

**Output:** Processed video  $V_{out}$  containing predicted rip current masks and bounding boxes.

Begin

For each frame  $F_i$  in  $V$  do

$F = \text{Backbone}(F_i)$

$S = \text{RegionScoring}(F)$

$R_K = \text{TopK}(S, K)$

$F_K = \text{FeatureSampling}(F, R_K)$

$Z = \text{TransformerEncoder}(F_K)$

$H = \text{TransformerDecoder}(Q, Z)$

$B = \text{BoundingBoxHead}(H)$

$C = \text{ClassificationHead}(H)$

$M = \text{SegmentationMaskHead}(H)$

For each prediction  $(B_i, C_i, M_i)$  do

If  $C_i = \text{rip current}$  and  $\text{score} > T_{\text{conf}}$  then

Overlay  $M_i$  on frame  $F_i$

Draw  $B_i$  on frame  $F_i$

End If

End For

Write processed frame  $F_i$  into  $V_{out}$

End For

Return  $V_{out}$

End

Figure 2 summarizes the RF-DETR data-preparation, dataset-construction, training, and evaluation workflow. Figure 3 provides the published architectural context. The source for the published results shown in Figure 3 is the original RF-DETR study [5].

### 3.2.2. YOLOv26

YOLOv26 is a one-stage segmentation model belonging to the YOLO family, which performs object detection and segmentation in a unified pipeline. The architecture consists of a backbone for feature extraction, a neck for multi-scale feature fusion, and a detection head for predicting bounding boxes and segmentation masks. YOLO-based models are widely used for real-time applications due to their efficiency and end-to-end design [21, 22]. In this study, multiple YOLOv26 variants are evaluated to analyze performance across different model scales. The model benefits from improved training strategies, including refined loss functions, better label assignment mechanisms, and more effective data augmentation techniques. These enhancements contribute to faster convergence and improved overall accuracy [14, 17].

To further highlight its advancements, YOLOv26 is compared with YOLO11, a recent previous version in the YOLO series. While YOLO11 already provides efficient real-time detection, YOLOv26 achieves better performance through enhanced feature extraction and improved multi-scale learning. These improvements lead to higher precision and recall, particularly in challenging scenarios such as small object detection and complex backgrounds. Moreover, YOLOv26 offers a better trade-off between speed and accuracy. It maintains real-time inference capability while delivering higher detection and segmentation performance compared to YOLO11 [19, 22]. Figure 4 presents the published accuracy-latency comparison between YOLOv26 and YOLO11, while Figure 5 summarizes the YOLOv26 data-preparation, training, and evaluation workflow.

Figure 2  
RF-DETR data-preparation, training, and evaluation workflow

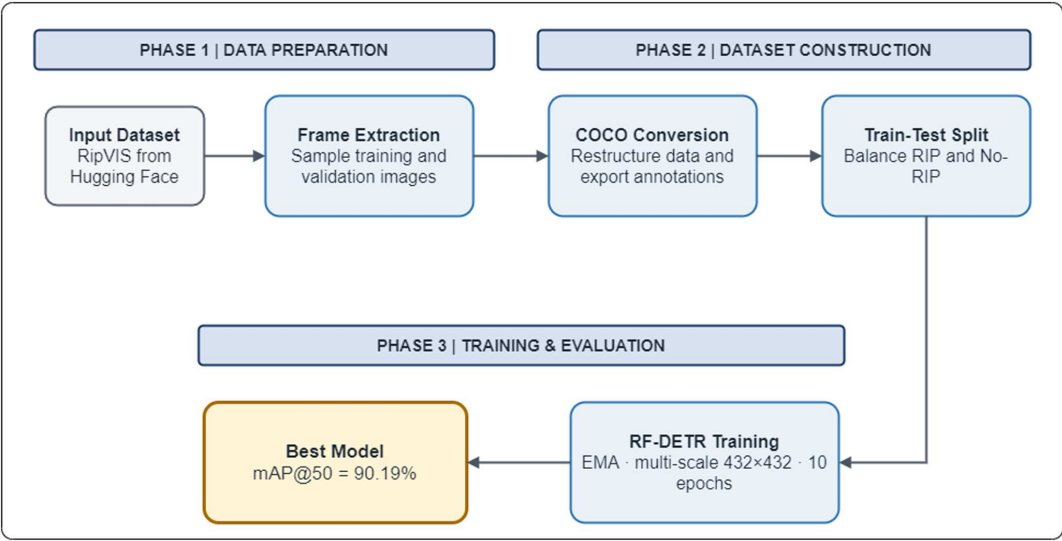


Figure 3  
Published RF-DETR accuracy–latency results reproduced for architectural context; compared method sources are reported in the original RF-DETR study

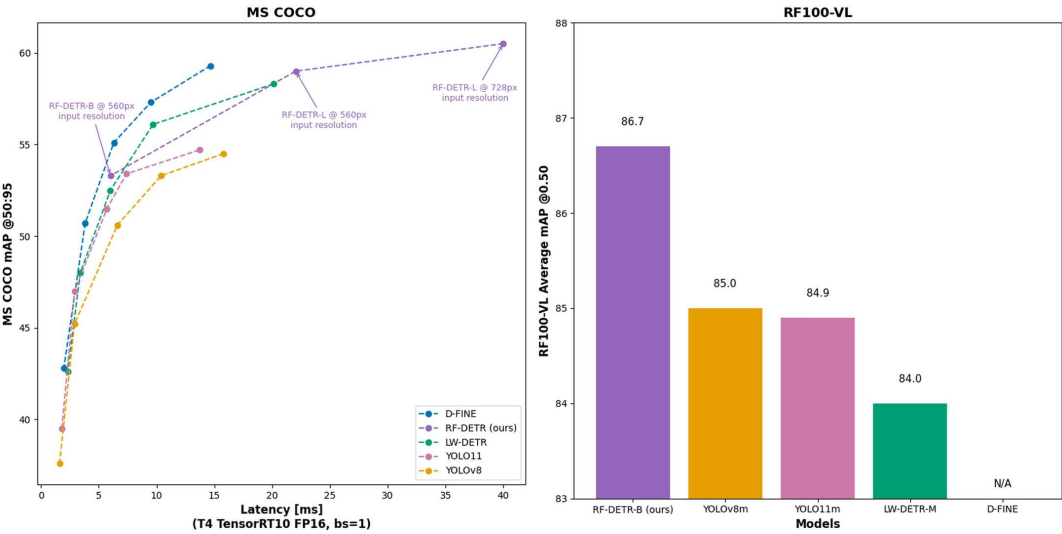
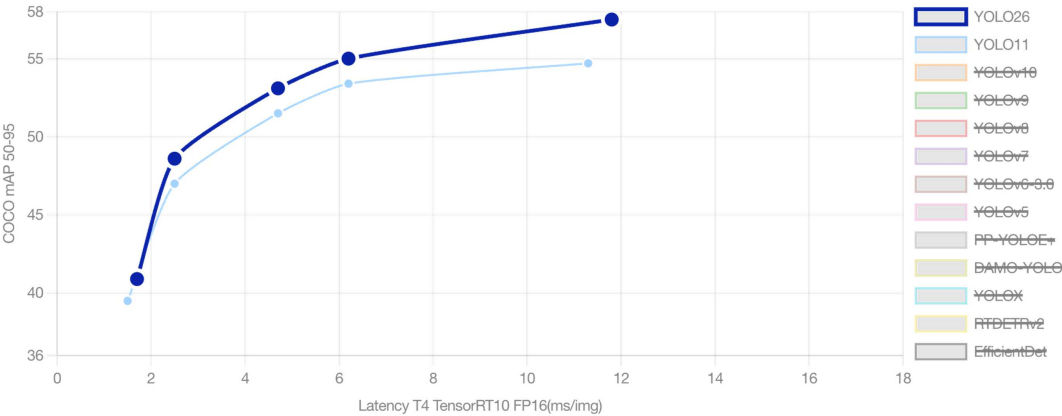
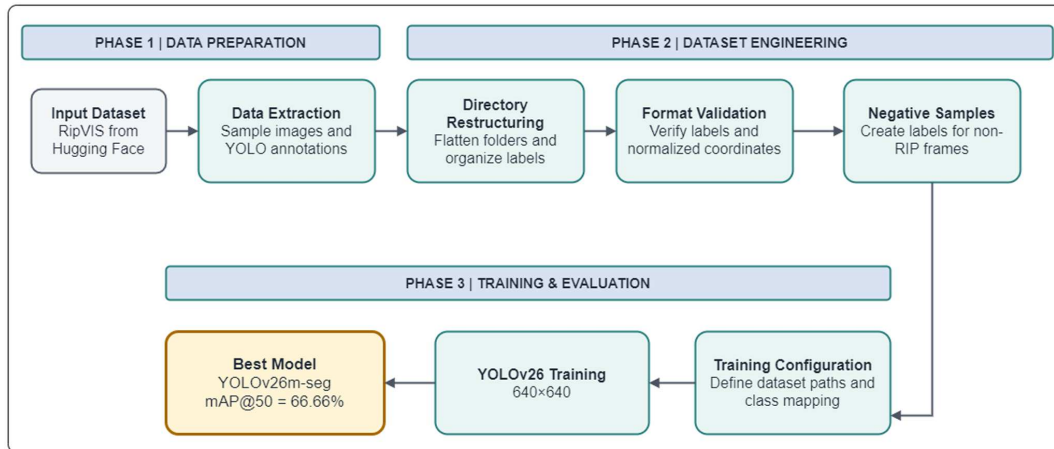


Figure 4  
Published accuracy–latency trade-off comparison between YOLOv26 and YOLO11 on COCO, reproduced for contextual discussion



**Figure 5**  
**YOLOv26 data-preparation, training, and evaluation workflow**



All YOLOv26 experiments used RipVIS in YOLO segmentation format and pretrained weights. The input size was  $640 \times 640$ , and mixed-precision training was enabled. The runs used the augmentation and mask-processing settings reported for their respective configurations. These settings describe the tested pipelines and were not selected through a controlled hyperparameter-sensitivity study.

The configurations differ in model variant, epoch count, batch size, and initial learning rate, as summarized in Table 3.

Because the configurations used different optimization settings, their results are configuration-specific and should not be interpreted as a controlled comparison of model scale alone.

Model performance is evaluated using precision, recall, F1-score, and mask mean average precision (mAP). The final quantitative comparison uses 4349 labeled evaluation images for each model. For RF-DETR, the final evaluation set combines the 2184-image training-time validation split and the 2165-image internal-test split. YOLOv26 uses the same 4349 labeled images directly as its evaluation/test set. Thus, the models share the final evaluation size but not the same training-time validation organization.

### 3.2.3. Results

The study evaluates RF-DETR and YOLOv26 for rip current detection and instance segmentation. Performance is reported using mask mAP@50 and mask mAP@50–95. The former uses an intersection over union (IoU) threshold of 0.5, whereas the latter averages performance over IoU thresholds from 0.5 to 0.95. Precision, recall, and F1-score are also reported where available. Training-loss components are used to describe convergence behavior.

Mask mAP evaluates the quality of predicted segmentation masks rather than bounding boxes. For RF-DETR, the reported loss components include classification loss, L1 bounding-box loss,

generalized intersection over union (GIoU) loss, mask cross-entropy loss, and Dice loss. These quantities describe optimization behavior and should not be treated as independent evidence of generalization.

RF-DETR was trained for 10 epochs through two consecutive five-epoch runs. Both standard and exponential moving average (EMA) evaluations are reported. YOLOv26 uses classification, bounding-box regression, and segmentation losses. The results describe the corresponding trained configurations; they do not isolate architectural effects from differences in optimization settings.

Information in Table 4 shows detailed training and evaluation results across all 10 epochs. The training process demonstrates a clear convergence trend. The total training loss decreased significantly from 46.80 at Epoch 1 to 35.13 at Epoch 10. The classification loss decreased from 4.65 to 3.21, the bounding box loss from 0.40 to 0.32, and the GIoU loss from 0.45 to 0.36. In addition, segmentation-related losses also improved, with mask cross-entropy loss decreasing from 0.80 to 0.65 and Dice loss from 1.02 to 0.81. These results indicate that the model progressively improves both object classification and segmentation quality during training.

In terms of segmentation performance, the model reaches its best regular performance at Epoch 5, achieving a mask mAP@50 of 0.8615 and mAP@50–95 of 0.3688. Beginning from Epoch 6, the model begins to stabilize and may approach saturation; performance fluctuates slightly despite continued loss reduction. With the EMA evaluation, the results exhibit more stable behavior across epochs. The best EMA mask mAP@50–95 is achieved at Epoch 8 with a value of 0.3646, while the highest EMA mask mAP@50 is observed at Epoch 10, reaching 0.9019. This indicates that EMA helps smooth parameter updates and provides more consistent generalization performance. Table 5 summarizes

**Table 3**  
**Configurations for YOLOv26 experiments**

| Model        | Epochs | Batch | Image size | LR    |
|--------------|--------|-------|------------|-------|
| YOLOv26n-seg | 50     | 32    | 640        | 0.010 |
| YOLOv26s-seg | 25     | 16    | 640        | 0.005 |
| YOLOv26m-seg | 20     | 24    | 640        | 0.003 |
| YOLOv26l-seg | 10     | 16    | 640        | 0.003 |

**Table 4**  
Detailed RF-DETR training and evaluation results over 10 epochs

| Epoch | Train loss | CE loss | BBox loss | GIoU loss | Mask CE | Mask dice | Regular mAP@50 | Regular mAP@50–95 | EMA mAP@50 | EMA mAP@50–95 |
|-------|------------|---------|-----------|-----------|---------|-----------|----------------|-------------------|------------|---------------|
| 1     | 46.80      | 4.65    | 0.40      | 0.45      | 0.80    | 1.02      | 0.7825         | 0.2978            | 0.8827     | 0.3562        |
| 2     | 41.43      | 4.00    | 0.36      | 0.39      | 0.74    | 0.89      | 0.8286         | 0.2917            | 0.9001     | 0.3401        |
| 3     | 38.83      | 3.69    | 0.34      | 0.38      | 0.70    | 0.86      | 0.8683         | 0.3416            | 0.8881     | 0.3511        |
| 4     | 38.36      | 3.62    | 0.34      | 0.37      | 0.70    | 0.85      | 0.8530         | 0.3385            | 0.8977     | 0.3622        |
| 5     | 36.22      | 3.38    | 0.33      | 0.37      | 0.67    | 0.83      | 0.8615         | 0.3688            | 0.8744     | 0.3636        |
| 6     | 37.21      | 3.45    | 0.35      | 0.38      | 0.68    | 0.85      | 0.8419         | 0.3215            | 0.8805     | 0.3561        |
| 7     | 37.28      | 3.41    | 0.34      | 0.37      | 0.70    | 0.84      | 0.8521         | 0.3359            | 0.8820     | 0.3510        |
| 8     | 36.93      | 3.39    | 0.33      | 0.37      | 0.68    | 0.83      | 0.8507         | 0.3300            | 0.8974     | 0.3646        |
| 9     | 35.96      | 3.30    | 0.33      | 0.36      | 0.67    | 0.82      | 0.8361         | 0.3323            | 0.8859     | 0.3519        |
| 10    | 35.13      | 3.21    | 0.32      | 0.36      | 0.65    | 0.81      | 0.8523         | 0.3286            | 0.9019     | 0.3479        |

**Table 5**  
Selected RF-DETR checkpoint results under different selection criteria

| Setting                     | Best Epoch | mAP@50 (M) | mAP@50–95 (M) | F1-Score |
|-----------------------------|------------|------------|---------------|----------|
| Regular (highest mAP@50)    | 3          | 0.8683     | 0.3416        | 0.8046   |
| Regular (highest mAP@50–95) | 5          | 0.8615     | 0.3688        | 0.8094   |
| EMA (highest mAP@50–95)     | 8          | 0.8974     | 0.3646        | 0.8506   |
| EMA (highest mAP@50)        | 10         | 0.9019     | 0.3479        | 0.8613   |

the best segmentation results observed during training. Under the standard evaluation setting, the highest mask mAP@50–95 is 0.3688 at Epoch 5. Under EMA evaluation, the best mask mAP@50–95 is 0.3646 at Epoch 8. Additionally, the highest EMA mask mAP@50 reaches 0.9019 at Epoch 10.

RF-DETR training was computationally intensive. The first phase required approximately 6 hours per epoch, while the second phase had shorter recorded durations because of differences in logging format; the training-time values should therefore be interpreted cautiously. Video inference used the selected EMA checkpoint and produced bounding boxes, confidence scores, and segmentation masks frame by frame. Red boxes and labels indicate detections, while semi-transparent green overlays show the predicted masks. Figure 6 summarizes the RF-DETR and YOLOv26 inference pipelines used in the video experiments.

We experimented on real-world rip current datasets executed on NVIDIA RTX A5000 GPU with a confidence threshold of 0.3 and mask transparency of 0.4. The first video, *DJI\_20260225\_0008.MP4*, contained 8494 frames and was processed in 14 min and 6 s, corresponding to an average inference speed of approximately 10.04 FPS. The second video, *DJI\_20260225\_0001.MP4*, contained 6037 frames and was completed in 12 min and 32 s, with an average speed of 8.03 FPS. The processed outputs were saved as *vid1\_output.mp4* and *vid2\_output.mp4*, respectively. Table 6 summarizes the RF-DETR inference settings and measured results for the two videos.

The RF-DETR outputs show that the fine-tuned model can produce rip current masks across the two evaluated UAV videos, with measured throughput of approximately 8–10 FPS on the reported hardware. These observations are limited to the two evaluated videos and should not be generalized to other lighting, tide, weather, season, viewing angle, or hardware conditions without further testing. The qualitative examples illustrate successful predictions, but a systematic failure-case analysis was not

available. Representative RF-DETR predictions for Videos 1 and 2 are shown in Figures 7 and 8, respectively.

The YOLOv26 experiments were conducted on four model scales: nano (n), small (s), medium (m), and large (l). All variants used 18,389 training images. Table 7 reports mask metrics on the 4349 labeled validation/evaluation images; these same 4349 images constitute the YOLOv26 final evaluation set. For RF-DETR, the corresponding final evaluation combines its 2184-image validation and 2165-image internal-test subsets.

Among the tested configurations, YOLOv26m-seg produced the highest reported mask mAP@50 (0.66662) and mask mAP@50–95 (0.29505). The other variants produced lower values under their respective training settings. Because the variants used different optimization schedules, these results do not isolate the effect of model scale.

In terms of training dynamics, all YOLOv26 models exhibited a consistent decrease in loss values, indicating stable convergence during training. For YOLOv26n-seg, the segmentation loss decreased from 2.66 to 0.99, while the classification loss dropped from 2.51 to 0.26 and the box loss from 1.68 to 0.69. Similar trends were observed for the small model, with segmentation loss decreasing from 2.52 to 1.29 and classification loss from 2.35 to 0.36.

YOLOv26m-seg showed decreasing training losses and the highest final evaluation metrics among the tested YOLOv26 configurations. This descriptive result applies only to the reported training schedule and does not establish that the medium model scale is inherently superior.

YOLOv26l-seg also showed decreasing losses but lower final evaluation metrics than YOLOv26m-seg under the reported settings. The shorter training schedule may have contributed to this outcome, but the present experiments do not establish a causal explanation.

Training time varied across the YOLOv26 configurations. The nano model required approximately 4.43 h for 50 epochs,

Figure 6  
RF-DETR and YOLOv26 inference pipelines

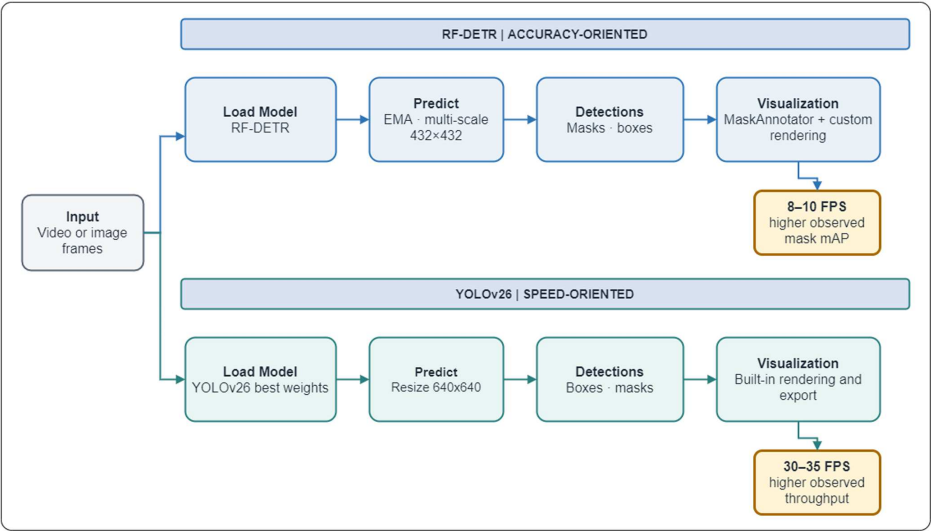


Table 6  
RF-DETR inference on two real-world RipVIS datasets

| Metric               | Video 1                 | Video 2                 |
|----------------------|-------------------------|-------------------------|
| Input video          | DJI_20260225_0008.MP4   | DJI_20260225_0001.MP4   |
| Output video         | rf-detr_vid1_output.mp4 | rf-detr_vid2_output.mp4 |
| Device               | CUDA                    | CUDA                    |
| Checkpoint           | Best EMA                | Best EMA                |
| Total frames         | 8494                    | 6037                    |
| Inference time       | 14 min 06 s             | 12 min 32 s             |
| Average speed        | 10.04 FPS               | 8.03 FPS                |
| Confidence threshold | 0.3                     | 0.3                     |
| Mask transparency    | 0.4                     | 0.4                     |

Figure 7  
Representative RF-DETR inference result on Video 1



**Figure 8**  
Representative RF-DETR inference result on Video 2



**Table 7**  
YOLOv26 results on the 4349-image labeled final evaluation set

| Model        | Best Epoch | mAP@50 (M) | mAP@50–95 (M) |
|--------------|------------|------------|---------------|
| YOLOv26n-seg | 25         | 0.62634    | 0.27435       |
| YOLOv26s-seg | 17         | 0.64879    | 0.27241       |
| YOLOv26m-seg | 19         | 0.66662    | 0.29505       |
| YOLOv26l-seg | 10         | 0.64173    | 0.26880       |

averaging 319 s per epoch. The small model required 2.64 h for 25 epochs (380 s per epoch), the medium model required 3.15 h for 20 epochs (568 s per epoch), and the large model required approximately 672 s per epoch. Because model scale, epoch count, batch size, and learning rate varied together, these observations do not establish a causal relationship between model capacity and performance. Video inference was evaluated frame by frame on two recordings, with detections rendered in the output videos and processing time recorded for each frame.

The first video, *DJI\_20260225\_0008*, contained 8494 frames. YOLOv26 completed inference successfully and saved the prediction result as *DJI\_20260225\_0008\_pred.mp4*. The average inference time per frame was 0.0299 s, corresponding to an estimated speed of 33.45 FPS. The second video, *DJI\_20260225\_0001*, contained 6037 frames. The model processed this video with an average inference time of 0.0286 s per frame and an estimated speed of 34.90 FPS, with the

output saved as *DJI\_20260225\_0001\_pred.mp4*. Table 8 summarizes the corresponding YOLOv26 inference results for both videos.

On the two evaluated UAV videos, YOLOv26 processed frames at the reported rate of more than 30 FPS on the experimental hardware. This limited result supports only a frame-wise efficiency observation for those recordings; it does not by itself establish real-time deployment suitability, long-duration robustness, or performance on resource-constrained platforms.

The two-video results indicate that YOLOv26 achieved higher frame-wise throughput than RF-DETR under the reported hardware and video settings. However, these observations are limited to two recordings and do not establish long-duration scalability, robustness under diverse environmental conditions, or readiness for safety-critical deployment. Representative YOLOv26 predictions for Videos 1 and 2 are shown in Figures 9 and 10, respectively.

**Table 8**  
YOLOv26 inference on two real-world RipVIS datasets

| Metric                       | Video 1                   | Video 2                   |
|------------------------------|---------------------------|---------------------------|
| Input video                  | DJI_20260225_0008.mp4     | DJI_20260225_0001.mp4     |
| Output video                 | yolov26_vid1_output.mp4   | yolov26_vid2_output.mp4   |
| Saved log                    | DJI_20260225_0008_log.txt | DJI_20260225_0001_log.txt |
| Total frames                 | 8494                      | 6037                      |
| Average inference time/frame | 0.0299 s                  | 0.0286 s                  |
| Estimated FPS                | 33.45                     | 34.90                     |
| Processing status            | Success                   | Success                   |

**Figure 9**  
YOLOv26 inference result on Video 1



**Figure 10**  
YOLOv26 inference result on Video 2



#### 4. Discussion

The experiments show an accuracy-throughput trade-off under the reported configurations. RF-DETR produced higher mask mAP values, whereas YOLOv26 achieved higher frame-wise throughput on the two UAV videos. The models used different pretrained weights, input resolutions, batch sizes, epoch counts, and optimization schedules. Accordingly, the comparison is configuration-specific and descriptive; it does not identify the cause of the observed performance difference.

In the reported runs, RF-DETR obtained higher mask mAP values than the YOLOv26 variants. However, the models were not trained under matched epoch counts, resolutions, batch sizes, or optimization schedules. Table 9 therefore summarizes configuration-specific results on the common 4349-image final evaluation set and should not be read as a controlled architectural benchmark or a statistically significant superiority claim.

Transformer attention and object queries may help represent long-range context in ambiguous marine scenes, but the present experiments do not isolate global attention, region selection, deformable attention, or any other component. The observed difference may also reflect the distinct pretrained models and training configurations. Consequently, the mechanism-level explanation remains a hypothesis for future ablation studies.

The measured accuracy-throughput difference must be interpreted within the unequal training configurations. RF-DETR produced higher mask mAP values but lower throughput, whereas YOLOv26 produced lower mask mAP values but higher throughput on the two videos. The study is limited to the tested setting and does not establish that either model is generally more suitable for safety-critical use; deployment choice would additionally require matched training, calibrated error costs, robustness testing, and platform-specific latency measurements. Table 10 provides a contextual comparison with methods reported in prior RipVIS work.

**Table 9**  
Comparison of selected segmentation results between YOLOv26 and RF-DETR based on mAP@50

| Model   | Setting/Variant      | mAP@50 (M) | mAP@50-95 (M) |
|---------|----------------------|------------|---------------|
| YOLOv26 | YOLOv26m-seg         | 0.6666     | 0.2951        |
| RF-DETR | EMA (highest mAP@50) | 0.9019     | 0.3479        |

**Table 10**  
Comparison with other methods on RipVIS dataset

| Models                    | mAP@50 | mAP@50-95 | FPS    |
|---------------------------|--------|-----------|--------|
| Mask R-CNN [1]            | 0.530  | –         | 7.84   |
| Cascade Mask R-CNN [1]    | 0.628  | –         | 9.53   |
| SparseInst R-50 [1]       | 0.703  | –         | 29.73  |
| SparseInst PVTv2 [1]      | 0.721  | –         | 27.99  |
| YOLO11n [1]               | 0.650  | –         | 128.20 |
| YOLO11s [1]               | 0.705  | –         | 116.27 |
| YOLO11m [1]               | 0.707  | –         | 76.93  |
| YOLO11l [1]               | 0.713  | –         | 57.14  |
| YOLO11x [1]               | 0.682  | –         | 34.01  |
| YOLOv26m-seg (this study) | 0.667  | 0.295     | 33.5   |
| RF-DETR (this study)      | 0.902  | 0.348     | 10.0   |

Within the reported YOLOv26 runs, the medium variant provided the strongest mask mAP values, while larger capacity did not improve the result under its shorter schedule. This observation is configuration-specific and should not be generalized to model scaling without controlled training-budget experiments. Likewise, the present data do not establish deployment suitability for either YOLOv26 or RF-DETR.

Table 10 places the present results beside values reported for methods evaluated in prior RipVIS work [1]. Because the publications may differ in preprocessing, implementation, hardware, training schedule, and evaluation partition, the table is contextual rather than a controlled ranking. The present RF-DETR and YOLOv26 entries are identified as this study; no claim of state-of-the-art performance or numerical improvement over the literature is made from this cross-publication comparison.

For the evaluated RipVIS setting, the results motivate further study of RF-DETR when mask accuracy is prioritized and YOLOv26 when frame-wise throughput is prioritized. A deployment recommendation would require failure-case analysis, matched training budgets, broader coastal conditions, and evaluation on the intended hardware.

**Limitations.** The study uses single reported training trajectories and does not include repeated trials, confidence intervals, statistical tests, exhaustive hyperparameter optimization, or component ablations. It does not benchmark MaskDINO, Video Mask2Former, SAM-based pipelines, or other recent transformer segmenters on the same split. Video inference is limited to two recordings, and systematic failure cases, floating-point operations (FLOPs), peak GPU memory, scalability tests, and embedded-platform latency are unavailable. These omissions limit causal interpretation, statistical generalization, robustness claims, and deployment conclusions.

## 5. Conclusion

This study evaluated two existing instance-segmentation models, RF-DETR and YOLOv26, for rip current detection and

segmentation on the RipVIS dataset. Both models were fine-tuned from pretrained weights using the same 18,389 training images and were finally evaluated on the same 4349 labeled images. In the RF-DETR workflow, these 4349 images were organized as a 2184-image validation split and a 2165-image internal-test split during training, while YOLOv26 used the 4349 labeled images as its validation/evaluation set. Therefore, the two models were compared using the same training-set size and final evaluation-set size, although their training-time validation organization was not identical.

Under the reported configurations, RF-DETR achieved higher mask mAP values, whereas YOLOv26 achieved higher frame-wise throughput on the two evaluated UAV videos. These findings suggest a practical accuracy-throughput trade-off for rip current segmentation: RF-DETR may be useful when mask accuracy is prioritized, while YOLOv26 may be useful when frame-wise processing speed is prioritized. At the same time, the comparison should be interpreted cautiously because the training schedules, input resolutions, batch sizes, and optimization settings were not matched across models.

The present results should therefore be understood as configuration-specific observations from an application study, not as evidence of a new model architecture, architectural superiority, enhanced spatial reasoning, or deployment readiness. The study also does not include repeated trials, statistical significance testing, component ablations, systematic hyperparameter search, broad environmental robustness evaluation, or embedded-platform benchmarking. Future work should conduct matched-budget experiments, repeated runs, component-level ablations, systematic failure-case analysis, computational profiling, and evaluation across diverse coastal conditions and target hardware.

## Acknowledgment

The authors would like to thank the authors of <https://huggingface.co/datasets/Irikos/RipVIS> contributors for sharing

data and the Department of Science and Technology of Quang Ngai Province for supporting the research team in conducting unmanned aerial vehicle (UAV) flights and image acquisition at My Khe Beach.

## Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

## Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

## Data Availability Statement

The data that support this study are openly available in Hugging Face at <https://huggingface.co/datasets/Irikos/RipVIS>, Video 1 at [https://drive.google.com/file/d/1fp1dJV0WGS9bGr2ZRbRksCHcfGEeSP\\_/view?usp=drive\\_link](https://drive.google.com/file/d/1fp1dJV0WGS9bGr2ZRbRksCHcfGEeSP_/view?usp=drive_link), Video 2 at [https://drive.google.com/file/d/15p2tU8HYv2lPau6uHFQ-RbXl0-R4C-sU/view?usp=drive\\_link](https://drive.google.com/file/d/15p2tU8HYv2lPau6uHFQ-RbXl0-R4C-sU/view?usp=drive_link), and Output video at [https://drive.google.com/drive/folders/1IfpnkifjXQ4B7n4gnkOR0P\\_O4AKjpa6?usp=drive\\_link](https://drive.google.com/drive/folders/1IfpnkifjXQ4B7n4gnkOR0P_O4AKjpa6?usp=drive_link).

## Author Contribution Statement

**Van Lam Ho:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Supervision, Project administration. **Van Khang Le:** Methodology, Software, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Xuan Vinh Le:** Supervision. **Trong Doi Nguyen:** Validation, Investigation, Resources, Data curation, Project administration. **Trang-Thi Ho:** Validation, Supervision.

## References

- [1] Dumitriu, A., Tatui, F., Miron, F., Ralhan, A., Ionescu, R. T., & Timofte, R. (2025). Ripvis: Rip currents video instance segmentation benchmark for beach monitoring and safety. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3427–3437. <https://doi.org/10.1109/CVPR52734.2025.00325>
- [2] Nguyex, X. L., & Dang, D. D. (2019). Assessing the possibility of appearing rip current at Quy Nhon Beach, Binh Dinh Province. *VNU Journal of Science: Earth and Environmental Sciences*, 35(4), 34–47. <https://doi.org/10.25073/2588-1094/vnuces.4416>
- [3] Dumitriu, A., Tatui, F., Miron, F., Ionescu, R. T., & Timofte, R. (2023). Rip current segmentation: A novel benchmark and YOLOv8 baseline results. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1261–1271. <https://doi.org/10.1109/CVPRW59228.2023.00133>
- [4] Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., & Xia, H. (2021). End-to-end video instance segmentation with transformers. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8737–8746. <https://doi.org/10.1109/CVPR46437.2021.00863>
- [5] Robinson, I., Robicheaux, P., Popov, M., Ramanan, D., & Peri, N. (2026). RF-DETR: Neural architecture search for real-time detection transformers. In *International Conference on Learning Representations*, 2026, 56087–56100.
- [6] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European Conference on Computer Vision*, 213–229. [https://doi.org/10.1007/978-3-030-58452-8\\_13](https://doi.org/10.1007/978-3-030-58452-8_13)
- [7] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2021). Deformable DETR: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 1–16.
- [8] Castelle, B., Dehez, J., Carayon, D., Liquet, S., & Savy, J. P. (2025). Simple physics-based rip current and shore-break wave hazard predictors for beaches in south-west. In *Coastal Dynamics 2025*, 41. [https://doi.org/10.1007/978-3-032-15473-6\\_36](https://doi.org/10.1007/978-3-032-15473-6_36),
- [9] Shafiei, H., Pearson, S. G., & Reniers, A. J. H. M. (2025). Analysis of rip-channel formation at an idealized beach in the framework of coastal sediment connectivity. In *Coastal Dynamics Conference*, 89–94. [https://doi.org/10.1007/978-3-032-15473-6\\_14](https://doi.org/10.1007/978-3-032-15473-6_14)
- [10] Brissman, E., Johnander, J., Danelljan, M., & Felsberg, M. (2023). Recurrent graph neural networks for video instance segmentation. *International Journal of Computer Vision*, 131(2), 471–495. <https://doi.org/10.1007/s11263-022-01703-8>
- [11] Yu, L., Tang, L., & Mu, L. (2025). A review of detection transformer: From basic architecture to advanced developments and visual perception applications. *Sensors*, 25(13), 3952. <https://doi.org/10.3390/s25133952>
- [12] Wang, Y., Zhang, X., Yang, T., & Sun, J. (2022). Anchor DETR: Query design for transformer-based detector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(3), 2567–2575. <https://doi.org/10.1609/aaai.v36i3.20158>
- [13] Li, F., Zhang, H., Liu, S., Guo, J., Ni, L. M., & Zhang, L. (2022). Dn-DETR: Accelerate DETR training by introducing query denoising. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4), 2239–2251. <https://doi.org/10.1109/TPAMI.2023.3335410>
- [14] Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., & Zhang, L. (2022). DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *International Conference on Learning Representations (ICLR)*, 1–20.
- [15] Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., & Girdhar, R. (2022). Masked-attention mask transformer for universal image segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1280–1289. <https://doi.org/10.1109/CVPR52688.2022.00135>
- [16] Ji, W., Li, J., Bi, Q., Liu, T., Li, W., & Cheng, L. (2024). Segment anything is not always perfect: An investigation of SAM on different real-world applications. *Machine Intelligence Research*, 21, 617–630. <https://doi.org/10.1007/s11633-023-1385-0>
- [17] Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7464–7475. <https://doi.org/10.1109/CVPR52729.2023.00721>
- [18] Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., & Ding, G. (2024). Yolov10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems*, 37, 107984–108011. <https://doi.org/10.52202/079017-3429>

- [19] Kühlechner, R. (2025). Object detection survey for industrial applications with focus on quality control. *Production Engineering*, 19(6), 1271–1291. <https://doi.org/10.1007/s11740-025-01369-4>
- [20] Çarklı Yavuz, B. (2026). Scale-dependent performance analysis of YOLO26 and YOLOv11 for PPE detection. *Electronics*, 15(6), 1146. <https://doi.org/10.3390/electronics15061146>
- [21] Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, 5(4), 1680–1716. <https://doi.org/10.3390/make5040083>
- [22] Stefenon, S. F., Matos-Carvalho, J. P., Mariani, V. C., dos Santos Coelho, L., & Yow, K. C. (2026). Diffusion-augmented YOLO26-Swin cascaded framework with hybrid SHAP-CAM for autonomous power grid inspection. *Autonomous Intelligent Systems*, 6(1), 13. <https://doi.org/10.1007/s43684-026-00135-2>
- [23] Castelle, B., Scott, T., Brander, R. W., & McCarroll, R. J. (2016). Rip current types, circulation and hazard. *Earth-Science Reviews*, 163, 1–21. <https://doi.org/10.1016/j.earscirev.2016.09.008>
- [24] Marqués-Alonso, A., Moreno, F. C., García-Maza, J. Á., Reyes-Merlo, M. Á., & Menendez-Telena, D. (2026). Rip current detection methods: A bibliometric perspective with coastal governance insights. *Ocean & Coastal Management*, 273, 107955. <https://doi.org/10.1016/j.ocecoaman.2025.107955>
- [25] Yadav, A., & Kumar, E. (2024). Object detection on real-time video with FPN and modified mask RCNN based on Inception-ResNetV2. *Wireless Personal Communications*, 138(4), 2065–2090. <https://doi.org/10.1007/s11277-024-11539-9>
- [26] Bianco, S., Buzzelli, M., Ciocca, G., Piccoli, F., & Schettini, R. (2026). A study on the generalization of DINOv2 features for food recognition tasks: A unified evaluation framework. *Intelligent Systems with Applications*, 29, 200632. <https://doi.org/10.1016/j.iswa.2026.200632>

**How to Cite:** Ho, V. L., Le, V. K., Le, X. V., Nguyen, T. D., & Ho, T.-T. (2026). Evaluating RF-DETR and YOLOv26 for Rip Current Detection and Segmentation. *Artificial Intelligence and Applications*. <https://doi.org/10.47852/bonviewAIA620210411>