RipVIS: Rip Currents Video Instance Segmentation Benchmark for Beach Monitoring and Safety

Supplementary Material

8. Overview

The supplementary material provides additional details and insights into the RipVIS dataset, experimental results, and methodology. While the main paper focuses on the major contributions and results, this document elaborates on the dataset's structure and diversity, the qualitative results of our experiments, and the impact of Temporal Confidence Aggregation (TCA) on rip current detection.

This supplementary aims to reinforce the robustness and reproducibility of our findings, offering a deeper understanding of the addressed challenges and proposed solutions. It also provides additional visualizations and metrics that could not be included in the main manuscript due to space limitations, including validation results (see Table 4).

RipVIS is a Video Instance Segmentation dataset, and it is challenging to convey its value in a static format. The supplementary material starts with a short description of the dataset variety in Section 9, with a visual showcase of all its diversity without masks, urging readers to see how many rip currents they can identify in Figure 6, before looking at the ground truths in Figure 7.

We continue in Section 10 with a deep dive into TCA, as exemplified in Figure 3. We describe the approach, its implementation methodology, its improvements and limitations, as well as final results. We also showcase TCA in action in more detailed scenarios, by sampling more frames from the same video. In Figure 5, TCA can be seen filtering false negatives, while in Figure 8, it can be seen filtering false positives, with a strong success rate, albeit not 100%. Lastly, we provide Figure 9, where TCA harms performance in a video transitioning from static to moving camera.

Finally, we finish with hyperparameter tuning in Section 11, diving deep into the hyperparameters that we used to train the different models, their strength, limitations and overall results. We analyze each model individually, discussing the approach used for hyperparameter tuning in each case. Ultimately, in Table 5, we present the standard deviations on all relevant metrics, for all models, on both validation and test sets.

9. Dataset Variety

Rip currents are complex, dynamic phenomena, requiring datasets that reflect their diversity in form, environment, and conditions. The RipVIS dataset was designed to capture this variety comprehensively, spanning different geographic locations, camera perspectives, and environmental scenarios.

Resolution	#Videos	FPS	#Videos
$4,096 \times 2,160$	1	60	14
$3,840 \times 2,160$	24	50	1
$2,730 \times 1,440$	1	30	119
$2,720 \times 1,530$	6	25	8
$2,560 \times 1,440$	2	24	8
$2,160 \times 3,840$	1		
$1,920 \times 1,080$	53		
$1,280 \times 720$	52		
$1,280 \times 676$	2		
$1,080 \times 1,920$	2		
$720 \times 1,280$	1		
480×360	3		
360×640	2		
Total	**150**	**Total**	**150**

Table 3. Resolution and FPS distribution of the 150 RipVIS videos containing rip currents, sorted by decreasing resolution and FPS. Videos are primarily landscape-oriented, with a few in portrait, reflecting real-world camera diversity. This variation enables robust evaluation across video qualities.

The dataset consists of 184 videos, totaling 212,328 frames. The videos are taken from multiple orientations and elevations, with different types of rip currents, in various weather conditions, from both seas and oceans. Figure 6 contains a large sampling from the videos, showcasing this variety, with Figure 7 showcasing their annotation masks. RipVIS videos are mainly in landscape orientation, with a few in portrait, reflecting real-world diversity in camera setups. For a detailed breakdown of the resolution and FPS distribution of RipVIS videos, see Table 3.

10. Temporal Confidence Aggregation (TCA)

TCA is an approach that enhances the consistency and reliability of rip current segmentation in video data by leveraging temporal information across consecutive frames. TCA effectively accumulates segmentation confidence over time, generating heatmaps that emphasize regions with stable rip current detections, while reducing noise from sporadic or transient detections.

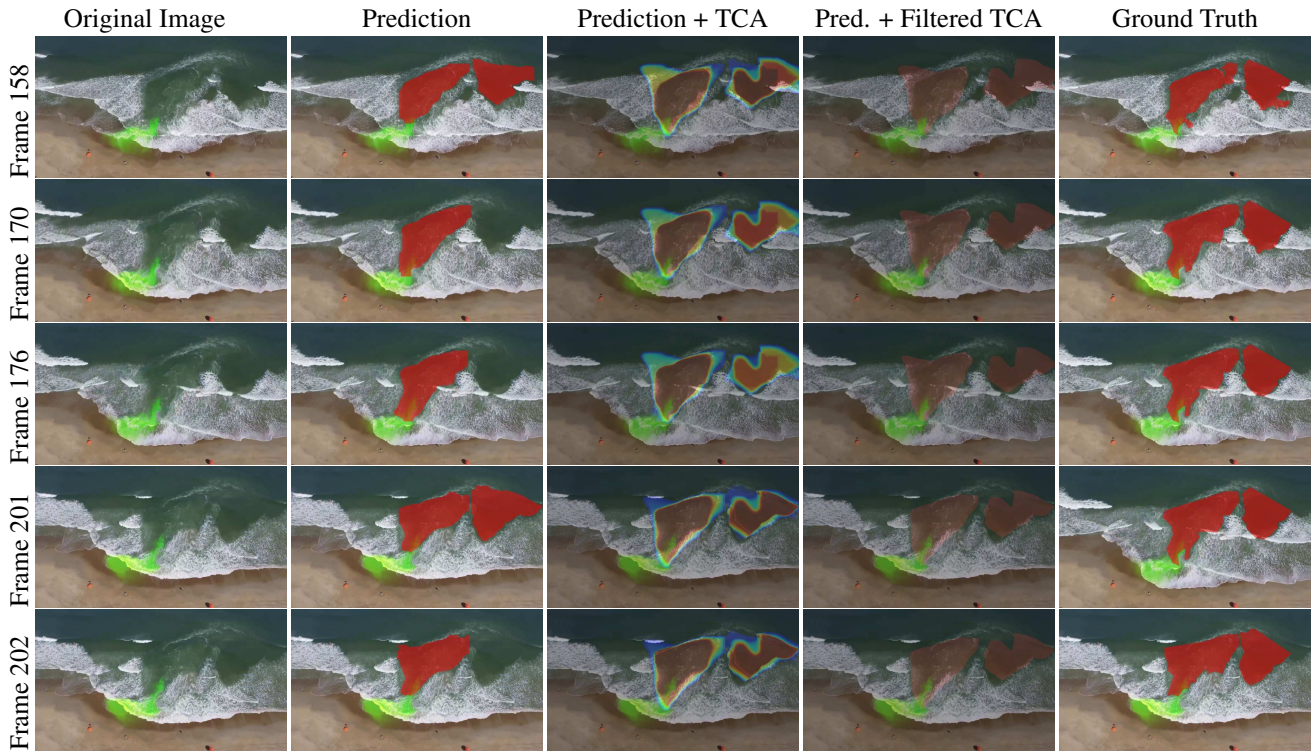

Figure 5. A more detailed example of TCA in action. All rows are of frames from the same video, showing how we mitigate for the false negative present in frames 176 (3rd row) and frame 202 (5th row).

10.1. Methodology

The TCA approach consists of several components that work together to aggregate segmentation confidence over time. Each component plays a role in dealing with the fluctuating and complex patterns of rip currents.

Heatmap initialization. For each instance, a heatmap is initialized as a two-dimensional array, where each value represents the accumulated segmentation confidence for a corresponding pixel in the video frame. This heatmap captures areas of high and consistent rip current activity, ensuring that these remain prominent throughout the analysis.

Heatmap update. The core of TCA lies in updating the heatmap over time by leveraging the current segmentation mask and information from previous frames. The confidence scores for each pixel are averaged across a short temporal window using the formula:

$$C_{avg}(t) = \alpha \cdot C(t) + (1 - \alpha) \cdot C_{avg}(t - 1),$$

where $C(t)$ is the confidence score at time t, $C_{avg}(t)$ is the aggregated confidence score, and α is the decay factor, set between 0 and 1, which dictates the influence of the current frame's confidence on the moving average. This step boosts the scores of consistently detected pixels. Additionally, every instance associated with a heatmap is accompanied by two supporting arrays:

Model	Precision	Recall	AP50	F_1	F_2
Mask-RCNN [24]	0.415	0.615	0.550	0.496	0.561
Cascade Mask-RCNN [5]	0.550	0.531	0.548	0.540	0.535
YOLO11n [28]	0.679	0.492	0.610	0.571	0.521
YOLO11s [28]	0.670	0.514	0.596	0.582	0.534
YOLO11m [28]	0.679	0.543	0.630	0.603	0.566
YOLO11l [28]	0.729	0.521	0.619	0.608	0.553
YOLO11x [28]	0.612	0.628	0.649	0.620	0.625
SparseInst R-50 [9]	0.477	0.664	0.564	0.555	0.615
SparseInst PVTv2 [9]	0.606	0.615	0.617	0.610	0.613

Table 4. Performance comparison of different models on the validation split. The models are applied on video and the metrics are calculated by evaluating on manually annotated frames. The best result on each metric is highlighted in blue.

- **Present counter:** This pixel-wise counter tracks the cumulative number of detections for each pixel within an instance's mask. Upon a detection, the counter increments for corresponding pixels, and growth is triggered only when the counter reaches a minimum threshold. This delay ensures that transient or spurious detections do not

prematurely inflate heatmap values.

- **Absence counter:** In contrast, this counter tracks the consecutive frames without a detection for each pixel. In the absence of a detection, the counter increases, triggering a reduction of heatmap values by a decay factor.

The heatmap update process is implemented using vectorized GPU operations, allowing efficient processing even for high-resolution video frames.

Heatmap smoothing. Rip currents often have amorphous shapes that change rapidly across frames. To maintain stability, while accommodating their fluid nature, a Gaussian smoothing filter is applied to the heatmap.

Hysteresis thresholding. TCA employs hysteresis thresholding to derive final binary masks from accumulated heatmaps, operating on the principle of differentiating strong and weak confidence scores within the heatmap. It uses an upper and a lower threshold. Pixels above the upper threshold are marked as strong detections, while those between the lower threshold and the upper thresholds form a weak detection. To connect these pixels, TCA applies a morphological dilation operation to each strong region, slightly expanding it to overlap with the weak mask. The final segmentation mask comprises strong pixels alongside weak pixels that are spatially connected to them.

Instance tracking. For each new frame, TCA tracks instances by matching them to IDs assigned in earlier frames.

10.2. Results and Discussion

The output of TCA is a heatmap that provides a confidence-weighted visualization of rip current segmentation over time. This aggregated heatmap is particularly beneficial for applications such as:

- **Rip current tracking:** Providing a stable representation of rip current activity, even when individual segmentations are noisy or inconsistent.
- **Beach safety monitoring:** Emphasizing regions of high rip current activity, which can help in developing early warning systems to alert beachgoers and lifeguards.

By aggregating temporal information, TCA effectively reduces the impact of sporadic false positives and false negatives, ensuring that only regions with consistent rip current activity are highlighted, making it a robust approach for rip current segmentation.

10.3. Limitations

While TCA provides significant improvements in the consistency of rip current segmentation, there are several limitations:

- **Increased computational requirements:** TCA requires maintaining and updating a heatmap in real-time, which can be computationally demanding, particularly for high-resolution video. Although GPU acceleration helps, substantial computational resources are still required.
- **Latency in highlighting rip currents:** Due to the need for multiple consistent segmentations before increasing confidence, TCA introduces some latency in highlighting newly detected rip currents. This can be a drawback for short videos or fast changing camera movement.
- **Parameter sensitivity:** The success of TCA hinges on well-adjusted parameters and thresholds. Consequently, although TCA can boost performance in tailored setups, achieving this becomes progressively more difficult as the setup broadens in scope.

11. Hyperparameter Tuning

This section provides an extended analysis of our experimental results, focusing on model performance on the RipVIS dataset and insights from hyperparameter tuning studies. The experiments are aimed to assess popular instance segmentation models for rip current detection and evaluate key hyperparameter impacts.

Most experiments are focused on varying backbones, optimizers, schedulers, and learning rates, as these hyperparameters greatly affect a model's ability to generalize and detect complex rip current patterns. Other parameters, like training epochs, early stopping patience, and batch size, were tested but showed minimal impact. To further enhance robustness, we extensively tested image augmentations for models implemented in Detectron2 (all except YOLO11, for which we used the built-in ones), exploring their effect on performance under diverse conditions.

In the following subsections, we provide a detailed description of the employed models, their configurations, and the conducted experiments. Each model was extensively evaluated under varying settings to identify the optimal configurations, understand their strengths and limitations, and assess their suitability for the challenging task of rip current segmentation in diverse video settings.

Mask R-CNN: Mask R-CNN [24], a two-stage model, extends Faster R-CNN with a segmentation branch, enabling simultaneous object detection and pixel-level masking. Using a Region Proposal Network (RPN) to generate Regions of Interest (RoIs), it excels at capturing irregular shapes like rip currents but sacrifices speed due to its complexity. In our tests, its performance was hampered by the dynamic nature of rip currents. For our experiments, we conducted an extensive study focusing primarily on different backbones, as these are critical for feature extraction. The backbones included ResNet-50-FPN [23], ResNet-101-FPN, ResNet-50-DC, and ResNet-101-DC, with FPN (Feature Pyramid Networks) enabling multi-scale feature extraction. Dilated Convolutions (DC), applied to specific stages of the backbone, expand the receptive field in these layers, enhancing spatial context capture for dense prediction tasks. In the experiments, we tested learning rates of 0.0025 and 0.005 with

Model	Validation Stddev					Test Stddev				
	Precision	Recall	AP50	F_1	F_2	Precision	Recall	AP50	F_1	F_2
Mask-RCNN [24]	0.06	0.09	0.07	0.07	0.08	0.05	0.08	0.07	0.06	0.07
Cascade Mask-RCNN [5]	0.05	0.08	0.07	0.06	0.07	0.06	0.07	0.06	0.06	0.07
YOLO11n [28]	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.03	0.03	0.04
YOLO11s [28]	0.03	0.03	0.03	0.03	0.03	0.02	0.04	0.03	0.03	0.04
YOLO11m [28]	0.04	0.03	0.03	0.04	0.03	0.05	0.04	0.03	0.03	0.04
YOLO11l [28]	0.06	0.04	0.04	0.03	0.04	0.04	0.03	0.03	0.04	0.03
YOLO11x [28]	0.04	0.04	0.03	0.04	0.04	0.05	0.04	0.04	0.04	0.04
SparseInst [9]	0.04	0.04	0.03	0.04	0.04	0.09	0.01	0.05	0.06	0.03

Table 5. Standard deviation summary for all models evaluated on the RipVIS dataset, with varied results across validation and test splits based on experiments.

Model	Train→Test	Accuracy
YOLO8n	Dumitriu *et al.* [16]→Dumitriu *et al.* [16]	0.750
	Dumitriu *et al.* [16]→RipVIS	0.205
YOLO11n	RipVIS→RipVIS	0.530
	RipVIS→Dumitriu *et al.* [16]	0.803

Table 6. Cross-dataset experiments on RipVIS vs. Dumitriu *et al.* [16] dataset.

the SGD optimizer and the Warmup Multi-Step LR scheduler.

Cascade Mask R-CNN: Cascade Mask R-CNN [5] builds on the Mask R-CNN architecture by introducing a multi-stage cascade of detectors and mask predictors, where each subsequent stage is trained to refine the outputs from the previous one with progressively stricter IoU thresholds. This cascading refinement process can enhance detection and segmentation accuracy, particularly for objects with complex or occluded boundaries. In principle, this approach is beneficial for segmenting ambiguous boundaries, such as those seen in rip currents, which often exhibit irregular and shifting patterns. While the multi-stage architecture helps mitigate false positives and improve instance mask quality, it does increase computational overhead. In practice, however, the model's performance on rip currents was limited, indicating potential challenges in handling highly amorphous and dynamic shapes.

Similar to Mask R-CNN, we conducted experiments focusing on backbone variations, using ResNet-50-FPN, ResNet-101-FPN, ResNet-50-DC, and ResNet-101-DC. Learning rates of 0.0025 and 0.005 were tested, alongside the SGD optimizer and Warmup Multi-Step LR scheduler.

YOLO11: In our experiments, YOLO11 [28] achieved reasonably high performance among the models tested for rip current segmentation, while also being the fastest. However, while it outperformed some models, it still struggled to accurately segment the complex rip current patterns present in our dataset, indicating that even advanced models like YOLO11 require further refinement to address the unique challenges of this task effectively. This performance highlights the difficulty of the problem and the need for continued work in developing specialized approaches for rip current detection.

For YOLO11, we performed the most extensive study, testing multiple configurations to maximize its performance. The study included all size variants (nano, small, medium, large, and x) and tested learning rates of 0.01 and 0.001, along with a weight decay of 0.0005. The models were trained using various optimizers, including SGD with momentum, Adam, AdamW, and standard SGD. The learning rate schedulers included both linear and cosine decay strategies.

We evaluated YOLO11 with both pre-trained weights and custom-trained weights, allowing us to analyze the impact of transfer learning on rip current detection. Pre-trained weights generally resulted in faster convergence and higher initial accuracy, while custom-trained weights offered more flexibility in adapting to the unique characteristics of the RipVIS dataset.

SparseInst: SparseInst [9] uses sparse instance activation maps for efficient, real-time segmentation, leveraging feature aggregation and bipartite matching to skip post-processing. This lightweight design minimizes computational overhead, making it ideal for dynamic tasks like rip current detection. We tuned it with ResNet-50, ResNet-101, and PVTv2 backbones, adjusting learning rates, optimizers (SGD, AdamW), batch sizes, and sparsity thresholds to balance sensitivity and noise. PVTv2 with data augmentation achieved the highest F_2 score among all models, alongside top F_1 and fast inference, making SparseInst the best overall choice for rip current detection.

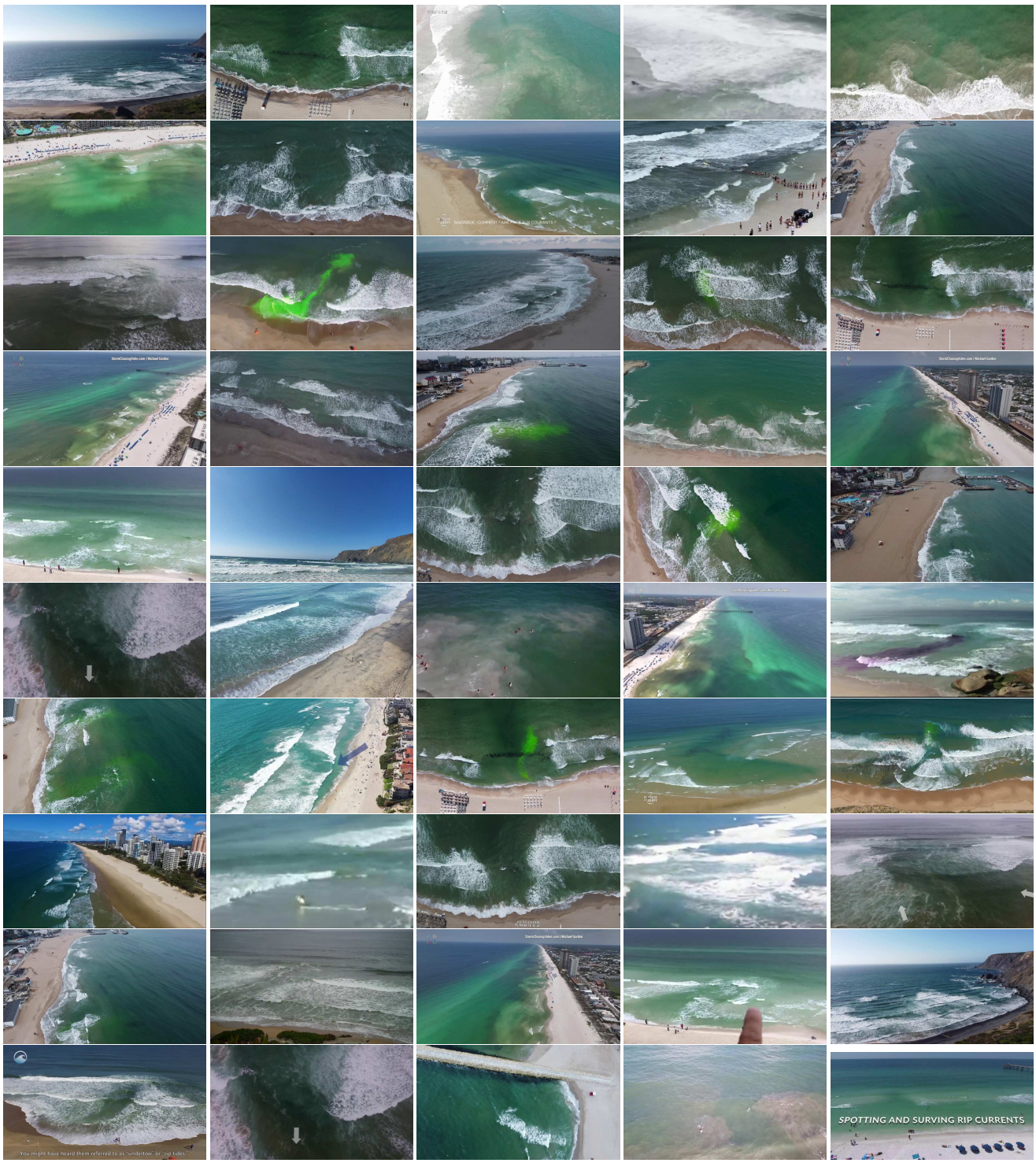

Figure 6. Examples of rip currents from the dataset, showcasing its diverse nature. Here we show frames from 55 randomly selected videos (out of 115 with rip currents). **Can you spot them all?** Some are easy, while others can be deceiving at first glance.

Figure 7. The same examples as before, with the ground truth masks overlayed on top. Pay special attention to the rip currents with sediments. How many did you get right?

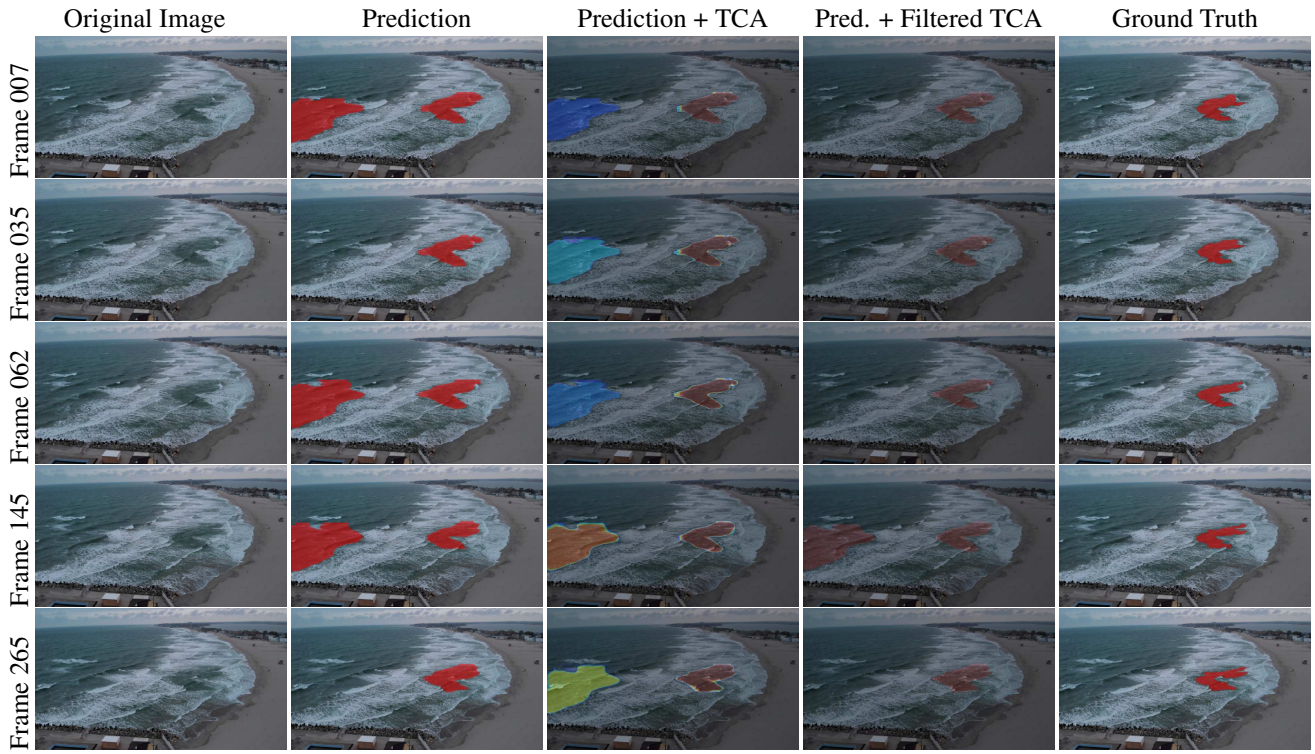

Figure 8. In this situation, TCA manages to filter many false positives, but not all. Too many false positives in a row get accumulated into a final detection (frames 062 - 145). Many false positives are on and off, though, and TCA helps filter most of them.

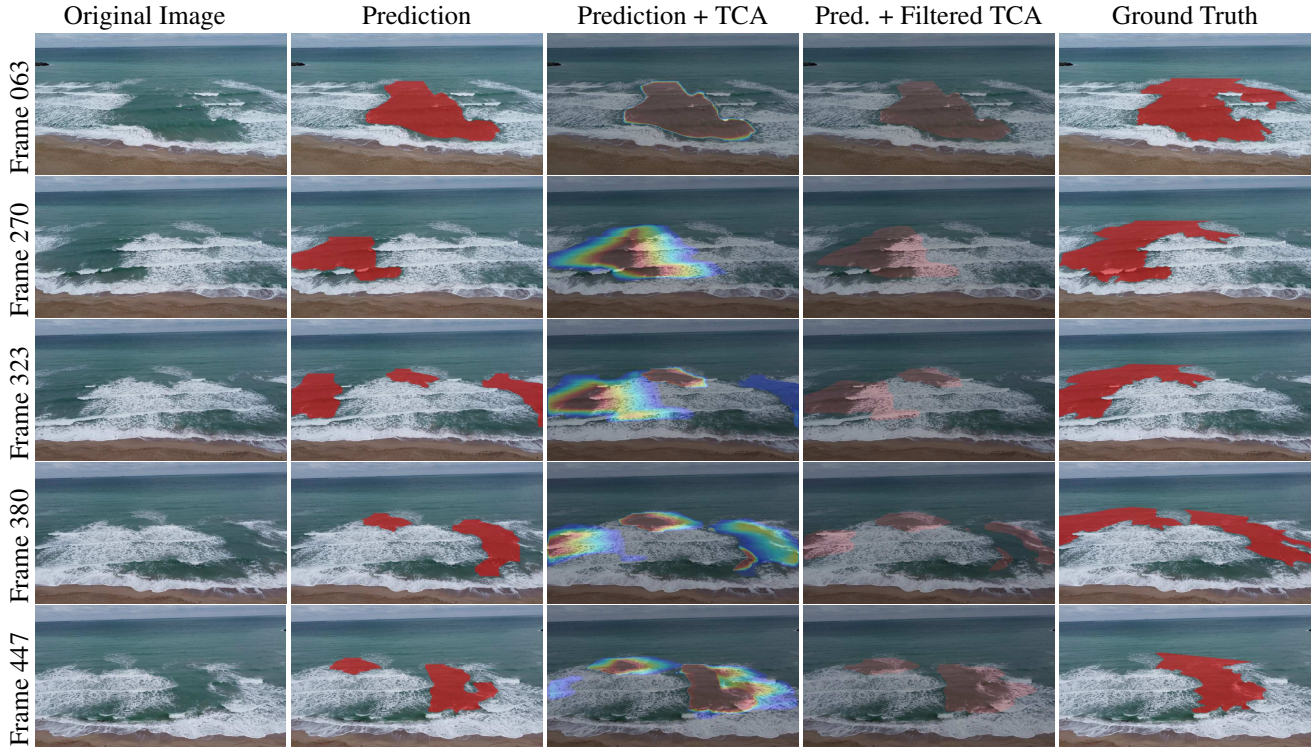

Figure 9. An example where TCA does more harm than good, if the camera is moving fast enough (in this case, the drone is dashing along the beachfront).