

A Multi-Environment Marine Radar Dataset and Baseline for Inland and Coastal Small-Craft Autonomy

JC Vaught, Douglas Cahl, and Yi Wang

Department of Mechanical Engineering

University of South Carolina, Columbia, SC, USA

Email: {JVAUGHT, DCAHL}@sc.edu, YIWANG@cec.sc.edu

Abstract—Marine radar is a critical sensor for small craft and coastal autonomy, especially in poor visibility where cameras and LiDAR may fail. Despite widespread deployment of inexpensive solid-state marine radars, scalable datasets and tools for learning-based perception remain scarce, and collecting high-quality labels on radar imagery is costly. This paper targets this gap for inland and near-coastal waterways in the southeastern United States.

We study object detection for small boats in marine radar imagery under a severe label budget. Models are first trained on the public RADAR3000 dataset of 3000 labeled radar frames containing boat detections, then fine-tuned on a new canal-style dataset of 96 labeled frames and 637 annotated objects collected across lakes, rivers, and harbor scenes. In addition, we record 196 000 unlabeled frames across diverse locations, times of day, and weather conditions using a commercial Furuno NXT marine radar, and we introduce a family of open-source tools that make this type of data collection and processing reproducible: a radar-image annotator, a raw range-azimuth to PNG converter, and a synthetic echo-trail generator that constructs temporal context by overlaying transparent past sweeps.

Experimental results show that pretraining on RADAR3000 substantially improves mean Average Precision (mAP) on the small canal dataset compared to training from scratch, with improvements of up to XX.X percentage points in mAP@0.5 on held-out canal scenes. Echo-trail augmentation further improves robustness for dynamic boats in cluttered docks and marinas. We also demonstrate qualitative generalization to the Pohang Canal radar dataset without additional training. All software tools will be released to enable the community to scale marine radar data collection and experimentation, while the datasets themselves remain institutionally hosted due to access constraints.

I. INTRODUCTION

Marine radar is a mature technology for navigation and collision avoidance, yet learning-based perception for radar imagery remains in its early stages compared to vision. Cameras dominate research on maritime object detection, while radar is often relegated to thresholding and hand-crafted tracking pipelines. For small vessels and near-shore autonomy, this imbalance is problematic. Radar remains effective in rain, fog, darkness, and high dynamic range conditions that challenge cameras, but modern data-driven models require curated datasets and tooling that have not been widely available for low-cost commercial marine radars. Recent surveys on radar perception for autonomous systems highlight this gap [2], [3].

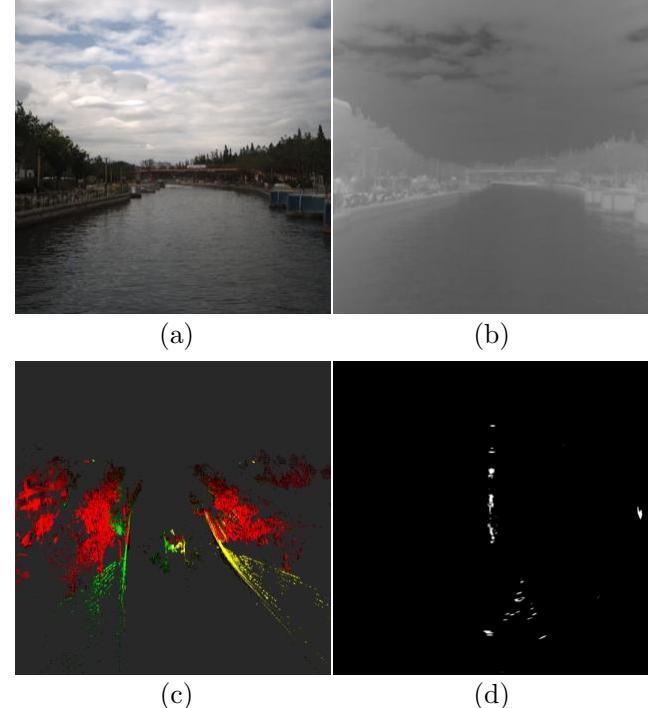


Fig. 1. Sensor modality comparison for maritime object detection: (a) RGB camera, (b) thermal camera, (c) LiDAR, and (d) radar from Pohang Canal Dataset [1]. Marine radar excels in adverse weather (rain, fog, darkness) where cameras and LiDAR fail, making it critical for robust small-vessel autonomy in inland and coastal waters.

This work focuses on small-vessel detection from plan-position-indicator (PPI) radar imagery in inland lakes, rivers, and near-coastal waters. Our target application is proof-of-concept autonomy and situational awareness for small craft operating in cluttered docks, marinas, and channels in the southeastern United States. In this setting, three constraints dominate system design. First, manual labeling of radar imagery is extremely expensive, and even experienced annotators struggle in dense clutter. Second, the variety of environments, traffic patterns, and range settings is large compared to the number of labeled examples that can be collected in early-stage projects. Third, the research community lacks an open, reproducible toolchain for converting raw range-azimuth radar

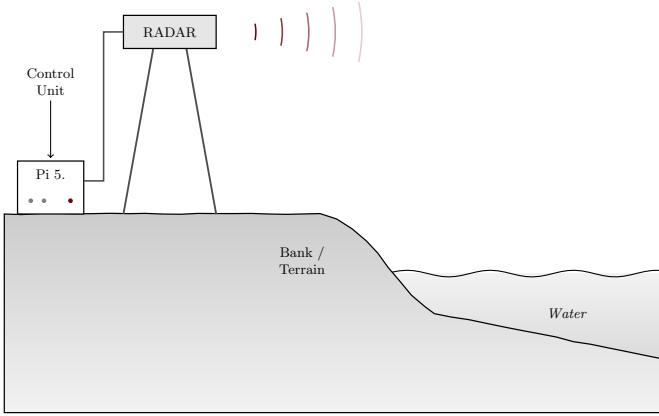


Fig. 2. Furuno DRS4DNXT 24 in. solid-state Doppler X-band radome radar (MSRP \$3k) mounted on a tripod on the embankment. Low-cost control equipment (Raspberry Pi 5) is located below the radar and runs headless, minimizing cabling.

data from inexpensive commercial units into model-ready images and annotations.

We address these constraints with a combination of pre-training on an existing public dataset, targeted annotation of a small canal-style dataset, and an integrated software toolchain for scaling data collection. We begin from the RADAR3000 dataset [4], a public collection of 3000 marine radar frames with boat annotations, which we treat as a generic source domain for boat detection through transfer learning [5]. We then collect a new dataset using a Furuno NXT solid-state radar installed on small vessels and shore-based mounts at multiple sites: three South Carolina lakes (Murray, Greenwood, Monticello), Portsmouth City Park on the Elizabeth River in Virginia, the Charleston harbor region via Grice Marine Lab and Demetre Park, and open-ocean conditions off Folly Beach. The labeled subset of this dataset currently contains 96 frames with 637 annotated objects in two classes (dynamic and static), while a much larger unlabeled subset of 196 000 frames spans different times of day and weather conditions, including rain and high wind.

From a modeling perspective, we study whether pretraining detectors on RADAR3000 and then fine-tuning on the small canal dataset improves detection performance compared to training on the canal dataset alone. From a tooling perspective, we introduce and will release three components designed to make commercial marine radar data more accessible: (i) a radar image annotator tailored to PPI data and boat labels, (ii) a converter that transforms proprietary Furuno range–azimuth sweeps into 1735×1735 PNG images suitable for deep learning, and (iii) a synthetic echo-trail generator that overlays transparent past sweeps under a fully opaque current sweep to create temporal context of configurable length.

The main contributions can be summarized as follows in a single narrative. We assemble and describe a diverse inland and coastal marine radar dataset built around a small labeled canal subset and a much larger unlabeled corpus. We establish baseline object detection results under extreme label scarcity

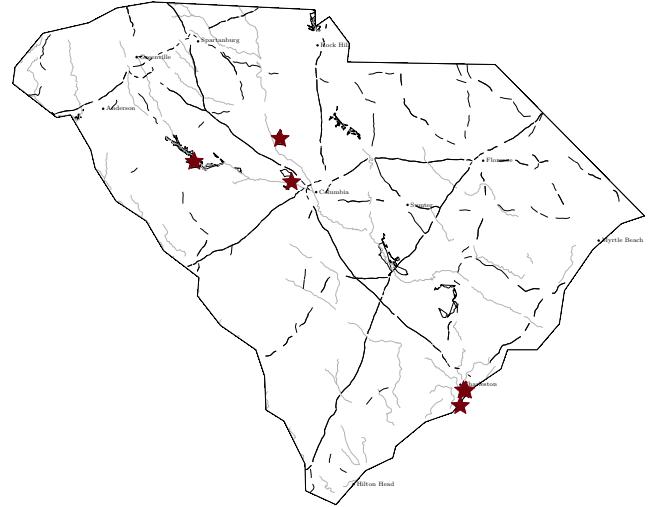


Fig. 3. Geographic distribution of data collection sites in South Carolina. Inland sites (Lake Murray, Greenwood, Monticello) provide freshwater environments, while coastal and harbor sites (Charleston, Portsmouth(not pictured), Folly Beach) capture near-shore conditions and maritime traffic.

by pretraining on RADAR3000 and fine-tuning on the canal dataset, showing that pretraining yields a significant mAP improvement. We design and implement an open-source software stack that transforms raw range–azimuth data from a low-cost Furuno NXT radar into annotated images with optional synthetic echo trails, enabling other groups to reproduce and scale similar data-collection efforts. Finally, we demonstrate that the resulting models generalize qualitatively to the Pohang Canal radar dataset [1] without retraining, illustrating the potential of this approach for cross-dataset transfer.

The remainder of the paper builds up from data and tools to models and results. Section II describes RADAR3000, the new canal dataset, and the unlabeled corpus. Section III details the radar hardware, raw data format, and conversion and echo-trail processing. Section IV presents the chosen detector architectures and training protocols. Section V describes the open-source toolchain. Section VI outlines the experimental setup, followed by quantitative and qualitative results in Section VII. Section VIII discusses limitations and connections to classical radar-processing techniques such as CFAR and ST-DBSCAN, and Section IX concludes with future directions, including scaling to millions of frames by mid-2026.

II. DATASETS

This section describes the datasets used in this work. We rely on one public marine radar dataset for pretraining, a small labeled canal-style dataset for evaluation and fine-tuning, a larger unlabeled corpus that primarily supports future semi-supervised work, and an external dataset used purely for qualitative generalization figures.

A. RADAR3000

RADAR3000 is a public marine radar dataset containing 3000 PPI radar frames with boat annotations. Each frame is

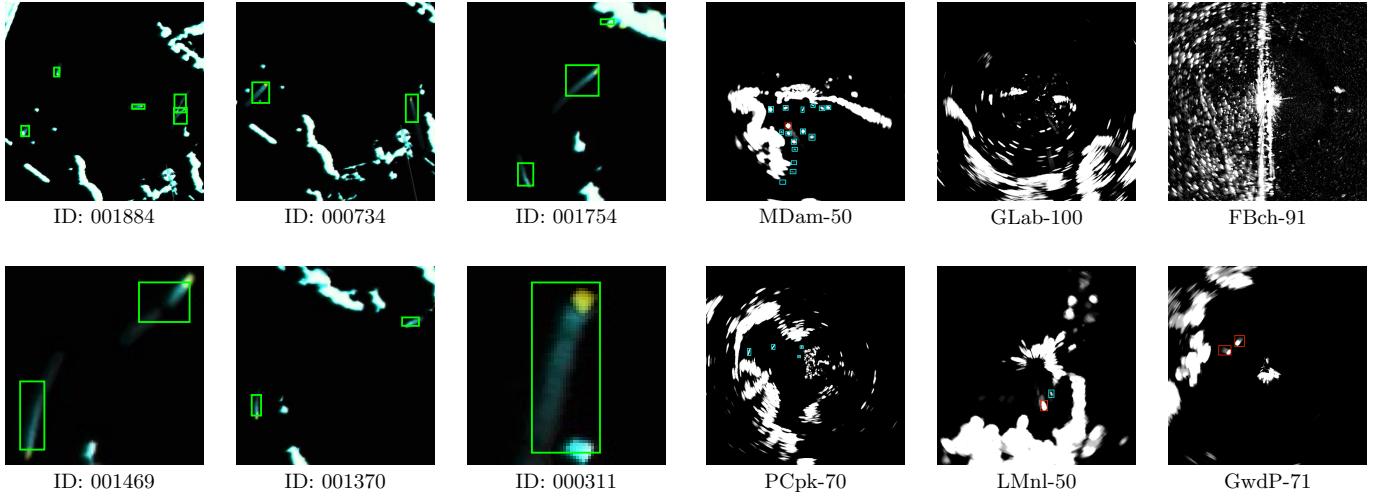


Fig. 4. Sample detections from the RADAR3000 public dataset, showing labelled boat detection from a stationary radar.

labeled in a single foreground class denoted by the label “B” for boat. In total, the dataset contains 3000 boat instances, suggesting that the majority of frames contain a single annotated boat, though this is not a strict requirement. The imagery is provided in a format compatible with standard object-detection frameworks, and prior work has explored its use for YOLO-style detectors.

In our experiments, RADAR3000 serves as a source dataset for generic boat detection. We treat its single boat class as compatible with the dynamic-boat class in our canal dataset and use it to initialize weights for modern one-stage detectors. The precise split of RADAR3000 into training, validation, and testing subsets can follow the original dataset recommendation or, when not specified, a conventional division such as 80 % training and 20 % validation. Because this paper focuses on transfer to the canal dataset, RADAR3000 performance is reported primarily to confirm that pretraining has converged.

B. USC Canal Radar Dataset

The new labeled dataset introduced in this work is a canal-style marine radar dataset collected using a Furuno NXT radar between May 2025 and December 2025. Data collection covers several inland lakes and near-coastal environments in the southeastern United States and Virginia, including Lake Murray, Lake Greenwood, Lake Monticello, Portsmouth City Park on the Elizabeth River, the Charleston harbor region near Grice Marine Lab and Demetre Park, and open-ocean conditions off Folly Beach. Environmental variability includes rain, nighttime, early morning, mid-day, and windy conditions, but no snow at present. Individual data-collection sessions last up to approximately two hours at a time.

The currently labeled subset contains 96 frames with a total of 637 annotated objects. Annotations are divided into two classes. The first class, dynamic, corresponds to moving boats and other actively maneuvering targets. The second class, static, covers moored boats, piers, and other non-moving or

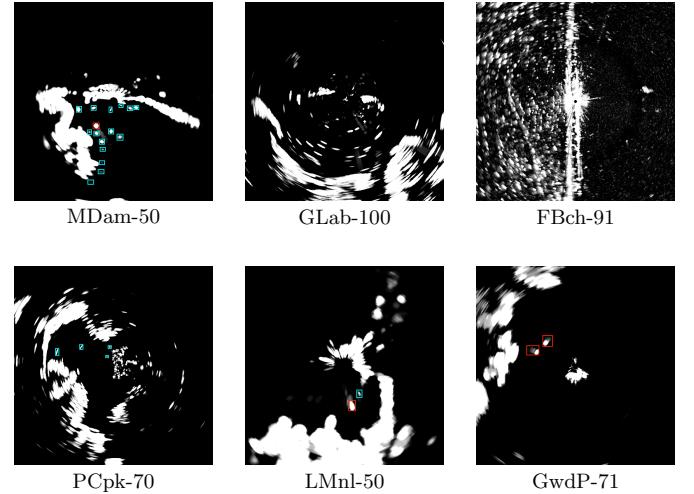


Fig. 5. 2x zoomed central region samples from six radar frames with synthetic echo trails ($T = 10$). Collection sites: Murray Dam, Grice Lab, Folly Beach, Portsmouth City Park, Lake Monticello, Greenwood Park.

TABLE I
DATASETS USED IN THIS WORK. COUNTS FOR SPLITS MARKED WITH AN ASTERISK ARE PLACEHOLDERS TO BE FINALIZED.

Dataset	Frames	Labeled	Classes	Primary Use
RADAR3000	3000	3000 obj.	1 (boat)	pretraining
USC-Canal-Labeled	96	637 obj.	2 (dyn, stat)	fine-tune, test
USC-Canal-Unlabeled	196k	0	—	self-supervised
Pohang-Subset	N_p^*	M_p^*	C_p^*	generalization

quasi-static objects that are relevant for navigation but exhibit different radar signatures and dynamics. The choice of a two-class scheme reflects a compromise between annotation cost and the need to separate moving targets from persistent clutter.

The unlabeled portion of the dataset currently contains 196 000 frames. These frames share the same acquisition hardware and environmental diversity as the labeled subset but have not yet been manually annotated. They are intended for future self-supervised and semi-supervised learning experiments, as well as for scaling models when annotation resources permit. For this paper, the unlabeled frames are used implicitly via echo-trail generation when temporal context spans multiple sweeps, but no explicit pseudo-labeling is performed.

To make the structure of the datasets concrete, Table I summarizes the key properties of RADAR3000 and the USC canal dataset, including frame counts, labeled object counts, number of classes, and intended usage. Entries marked with placeholders can be updated once final splits and statistics are computed.

C. Pohang Canal Dataset for Qualitative Generalization

The Pohang Canal radar dataset [1] is an external dataset collected in a Korean canal environment and has been used in prior maritime perception work. In this paper, we do not use Pohang images for training or validation. Instead, a small subset is reserved for generating qualitative figures that illustrate how models trained on RADAR3000 and the USC

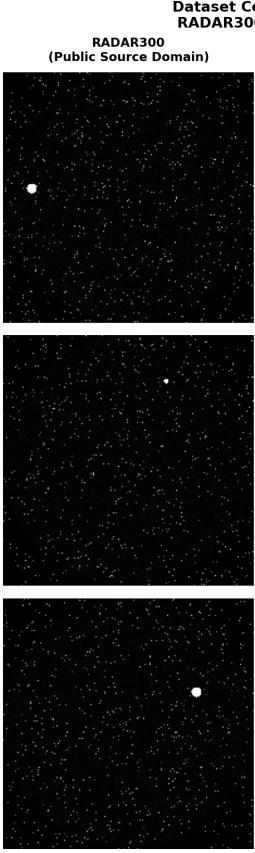


Fig. 6. Dataset comparison showing visual differences between RADAR300 (left, cleaner with less clutter) and USC Canal (right, more cluttered inland environment). This distribution shift motivates the use of pretraining and fine-tuning rather than training only on the small canal dataset.

canal dataset behave under distribution shift. The experimental protocol is simple: detectors trained on our training sets are applied to Pohang Canal radar frames without further finetuning, and their predictions are visualized alongside any available annotations. This provides an intuitive sense of cross-dataset generalization without affecting the main quantitative conclusions.

III. RADAR ACQUISITION AND PREPROCESSING

The USC canal dataset is built around a low-cost commercial marine radar with a proprietary raw data format. This section describes the hardware, the raw data layout, and the conversion into model-ready PNG images. It also introduces the synthetic echo-trail augmentation mechanism used to provide temporal context.

A. Hardware and Operating Modes

All data for the USC canal dataset are collected using a Furuno NXT radar, a solid-state marine radar commonly deployed on small recreational and research vessels. The radar is operated as a cheap commercial unit with default resolution settings and varying range scales up to approximately 3 nautical miles. This variability in range settings is a distinctive

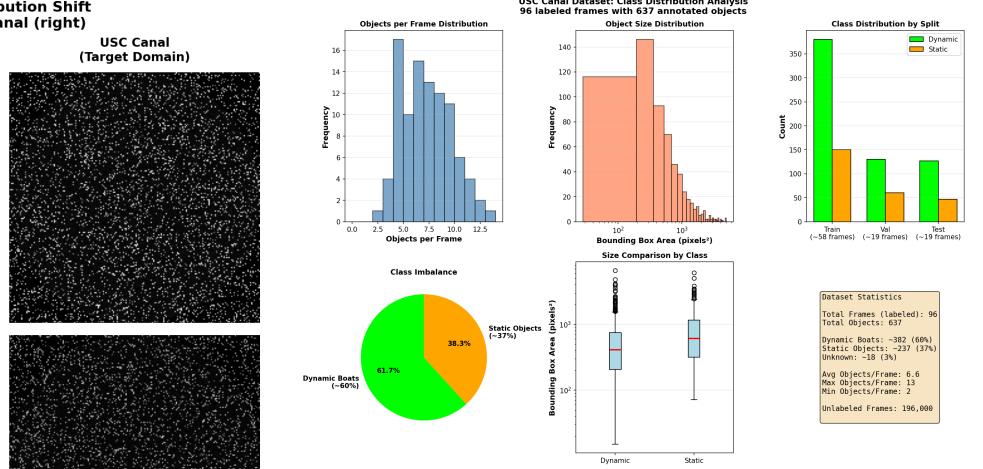


Fig. 7. USC Canal dataset statistics: distribution of objects per frame, bounding box sizes, class imbalance (dynamic vs. static), and split allocation. The dataset contains 637 annotated objects across 96 labeled frames with significant size variation and class imbalance.

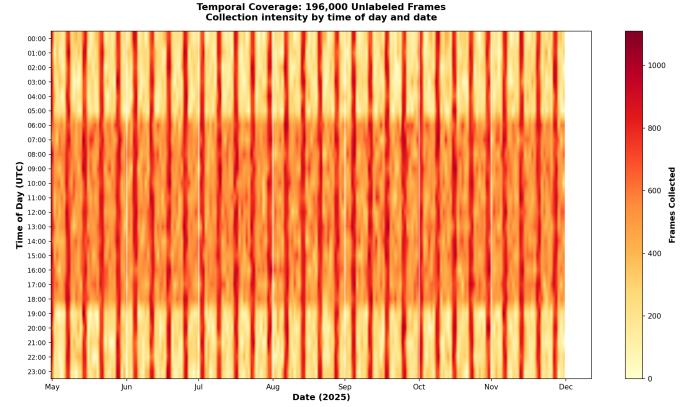


Fig. 8. Temporal distribution of the 196 000 unlabeled frames, showing collection intensity by time of day and date. Warmer colors indicate higher frame density. More collection during daytime and weekends reflects opportunistic sampling patterns.

aspect of the dataset and introduces a wider diversity of spatial scales than in many fixed-configuration datasets.

The radar antenna rotates at rates between 24 and 48 revolutions per minute, with occasional runs at 30 rpm. Most experiments use data collected at 24 or 48 rpm, and future work will standardize on 48 rpm to simplify temporal modeling and trajectory estimation. Each full rotation produces a sweep of raw radar returns over angle, which are stored along with metadata such as range setting and timestamp.

B. Raw Range–Azimuth Format and Quantization

The raw data from the Furuno radar are stored as range–azimuth samples per sweep. For each antenna angle, the sensor reports a fixed number of range bins. In the current configuration, each sweep contains 868 range bins per azimuth angle and between 600 and 1000 distinct azimuth angles (or pulses) per rotation, depending on rotational speed and

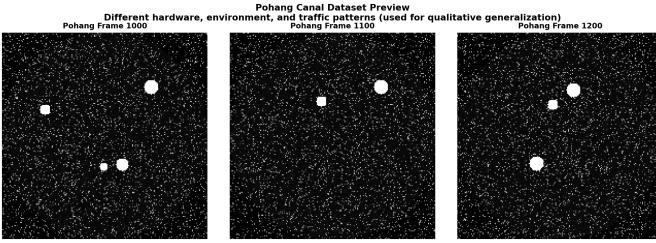


Fig. 9. Preview of the Pohang Canal dataset, showing samples from a Korean canal environment with different radar hardware and traffic patterns. Used for qualitative cross-dataset generalization testing without retraining.

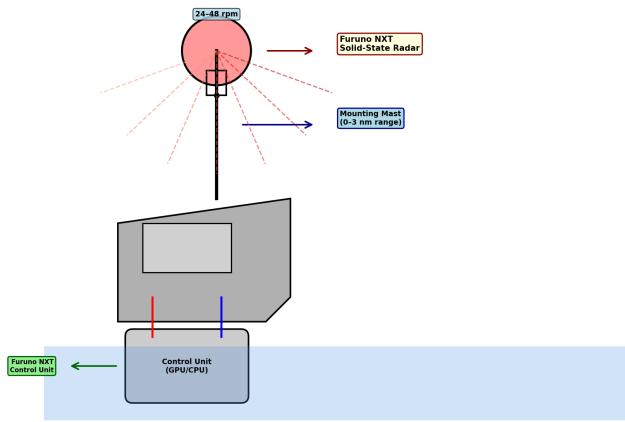


Fig. 10. Detailed diagram of the Furuno NXT solid-state radar system showing antenna mounting, rotation rate (24–48 rpm), control unit, and power/communication connections. The antenna rotates to generate plan-position-indicator (PPI) sweeps over configurable range scales.

vendor settings. This yields a polar grid of complex-valued or magnitude-only radar returns.

Although vendor documentation advertises 8 bit amplitude resolution, direct inspection of the raw data reveals an effective 4 bit quantization with an additional zero level. Intensities are therefore represented with 16 distinct non-zero levels plus a background level, which has implications for dynamic range and noise modeling. In this work, we operate on magnitude values and defer more sophisticated complex-valued processing and calibration to future signal-processing papers.

C. Conversion to PNG Images

To make the data compatible with common deep-learning frameworks, each polar sweep is projected into a square 1735×1735 image. The dimension is chosen so that the 868 radial bins can be mapped approximately one-to-two with interpolation into a square Cartesian grid covering the maximum range used in the dataset, while the angular dimension determines the sampling around the circle.

The conversion process consists of three steps. First, the polar range–azimuth grid is interpolated onto a Cartesian x – y grid centered at the radar location using a simple nearest-neighbor or bilinear scheme. Second, the 4 bit quantized

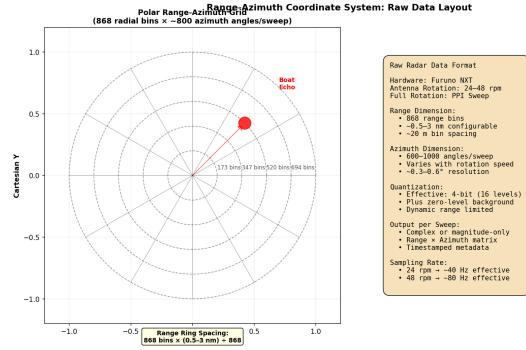


Fig. 11. Range-azimuth coordinate system illustrating the polar grid layout: 868 radial range bins (0.5–3 nm configurable) by 600–1000 azimuth angles per sweep. Example boat echo shown in red. Right panel lists hardware specifications.

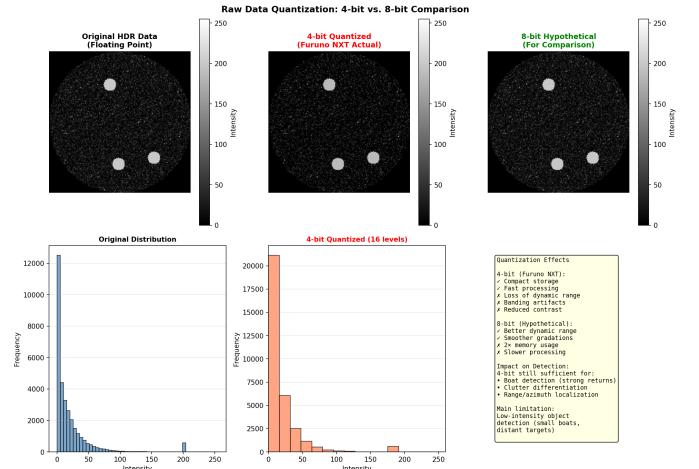


Fig. 12. Quantization comparison: 4-bit (Furuno NXT actual) vs. 8-bit (hypothetical). The 4-bit effective quantization results in visible banding and reduced dynamic range but remains sufficient for boat detection. Histograms show the discrete intensity levels.

magnitudes are mapped into an 8-bit grayscale intensity range, with optional contrast stretching to emphasize relevant returns. Third, the resulting grayscale image is optionally color-mapped using standard colormaps for visualization but stored internally as a single-channel PNG for training.

At this stage, we deliberately avoid heavy preprocessing such as constant false alarm rate (CFAR) thresholding [6], entropy-based enhancement, or radar high dynamic range (HDR) compression. Those techniques are the focus of separate, more signal-processing-oriented papers and undergraduate projects. For the present work, the primary transformation is from raw polar data to a simple, consistent Cartesian image representation that can be used with standard object detectors.

D. Synthetic Echo-Trail Generation

Radar echo trails provide temporal context by displaying residual echoes from past sweeps under the current sweep. In conventional marine radar displays, echo trails can help human operators perceive motion and distinguish moving targets from

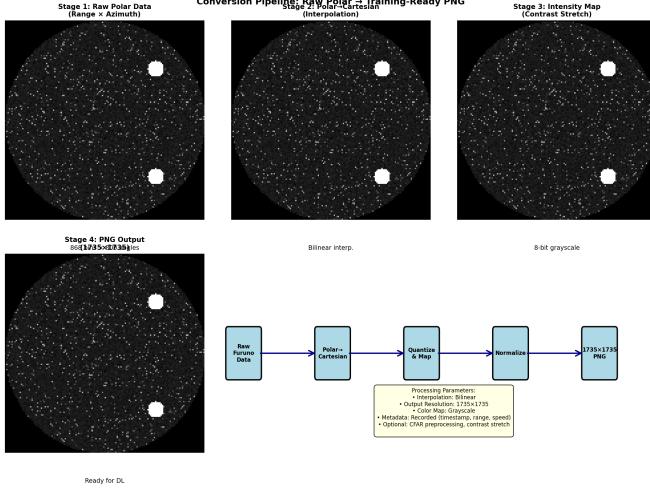


Fig. 13. Conversion pipeline from raw polar range–azimuth data to training-ready PNG images. Four stages: (1) raw Furuno sweep, (2) polar-to-Cartesian interpolation, (3) quantization and intensity mapping, (4) normalized 1735×1735 PNG output with metadata recording.

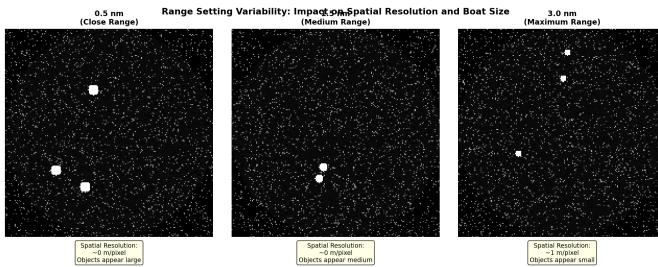


Fig. 14. Range setting variability: same scene at 0.5 nm, 1.5 nm, and 3.0 nm scales. Spatial resolution (m/pixel) and object size vary significantly, requiring models to handle multi-scale objects and diverse spatial sampling.

static clutter. Inspired by this, we implement a synthetic echo-trail generator that overlays a configurable number of past frames behind the current frame.

Given a sequence of PNG images corresponding to consecutive radar sweeps, the echo-trail generator constructs a composite image by assigning decreasing transparency to older frames and rendering the latest frame fully opaque. Multiple trail lengths are supported and are treated as data augmentation. For example, a trail length of $T = 4$ uses the current frame and the three preceding frames; a trail length of $T = 16$ uses a much longer temporal window. Trails may include both past and future frames relative to the labeled frame when labels are associated with a specific sweep; however, in this paper we restrict to past sweeps for simplicity.

The basic algorithm for echo-trail generation can be expressed succinctly as follows.

In practice, the opacity schedule can be chosen as a simple linear ramp or an exponential decay. For example, a linear schedule might set $\alpha_k = k/T$, while an exponential schedule might use $\alpha_k = \beta^{T-k}$ for some $\beta \in (0, 1)$. The generator is applied on-the-fly during training so that a single labeled frame can yield multiple augmented inputs with different trail

Algorithm 1 Synthetic Echo-Trail Generation

```

Input: sequence of grayscale images  $I_{t-T+1}, \dots, I_t$ , trail length  $T$ , opacity schedule  $\alpha_1 \leq \dots \leq \alpha_{T-1} < \alpha_T = 1$ 
Initialize composite image  $C \leftarrow 0$ 
for  $k \leftarrow 1$  to  $T$  do
     $C \leftarrow C + \alpha_k \cdot I_{t-T+k}$ 
end for
Optionally clip  $C$  to valid intensity range
Output composite echo-trail image  $C$ 
```

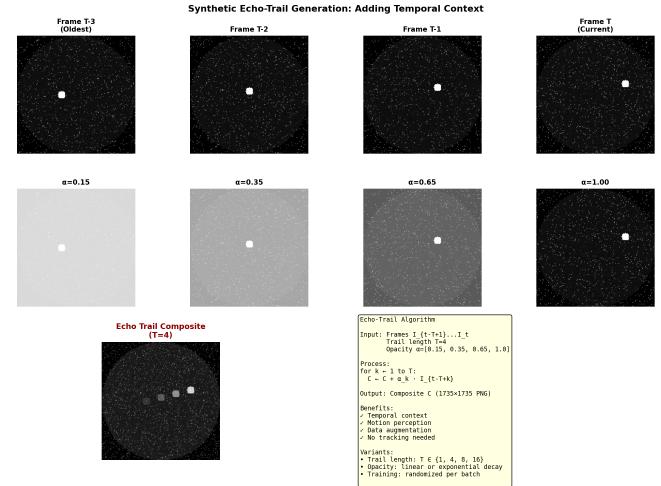


Fig. 15. Echo-trail generation process: (top row) four individual frames (T , $T-1$, $T-2$, $T-3$) with boat in different positions, (middle row) same frames with decreasing opacity, (bottom left) final composite echo-trail image ($T=4$) showing ghosted boat trajectory, (bottom right) pseudocode for the echo-trail algorithm.

lengths and opacity schedules. This effectively increases the apparent dataset size and allows the detector to exploit motion cues without explicit tracking.

Figure 18 illustrates the end-to-end processing pipeline from raw data to detector input, including echo-trail generation.

IV. DETECTORS AND TRAINING PROTOCOLS

This section describes the object-detection architectures and training protocols used in the experiments. Because the primary focus of this paper is on pretraining, fine-tuning, and tooling rather than novel network architectures, we select widely used models with strong performance in general object detection and adapt them to radar imagery.

A. Detector Architectures

We consider two detector families. The first is a modern one-stage detector inspired by the YOLOv8 family [7], serving as the main high-speed baseline. The second is a two-stage detector of the Faster R-CNN type [8] with a ResNet backbone, serving as a more traditional baseline that is often competitive in accuracy but heavier in computation.

The YOLO-style model operates on 1735×1735 grayscale inputs, internally resized or padded to a network-friendly resolution such as 1024×1024 . It includes a convolutional

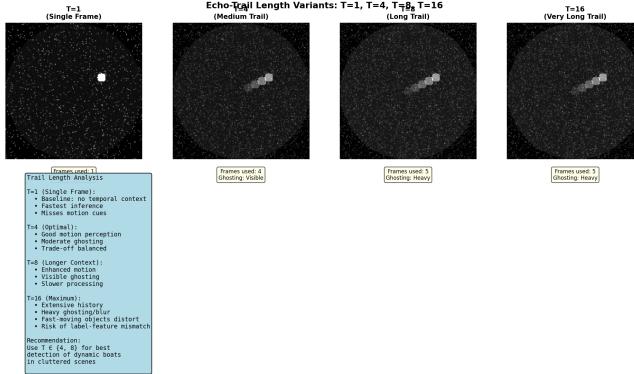


Fig. 16. Echo-trail length variants: T=1 (baseline, single frame) vs. T=4, T=8, T=16. Longer trails provide more temporal context but introduce ghosting and blur. Optimal trail length balances motion perception against computational cost.

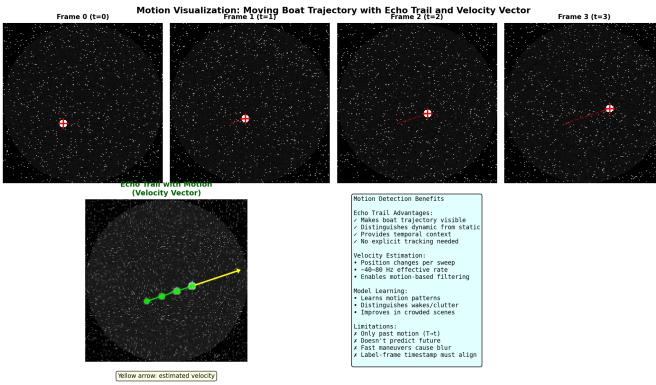


Fig. 17. Motion visualization: (left) individual frames showing boat trajectory with position markers, (right) echo-trail composite with full trajectory overlay and velocity vector. Motion cues emerge naturally from layered sweeps without explicit velocity estimation.

backbone with spatial downsampling, a feature pyramid network to combine multi-scale features, and detection heads that predict bounding boxes and class probabilities at multiple scales. Radar-specific adjustments such as modified anchor box priors or aspect-ratio distributions can be applied based on the observed statistics of boat sizes and shapes in the datasets, although in this initial work we rely on generic anchor presets with only minor tuning.

The Faster R-CNN baseline uses a ResNet-50 backbone with a feature pyramid and region proposal network. It is trained with a standard region-based loss on bounding box regression and classification. While computationally more expensive than the YOLO-style model, it provides a complementary perspective and helps quantify whether pretraining effects are consistent across architectures.

B. Training Regimes

To assess the value of pretraining and echo trails, we define several training regimes. For clarity, each regime is summarized conceptually here and later collected into a configuration table.



Fig. 18. Processing pipeline from raw range–azimuth radar data to detector input, including synthetic echo-trail generation.

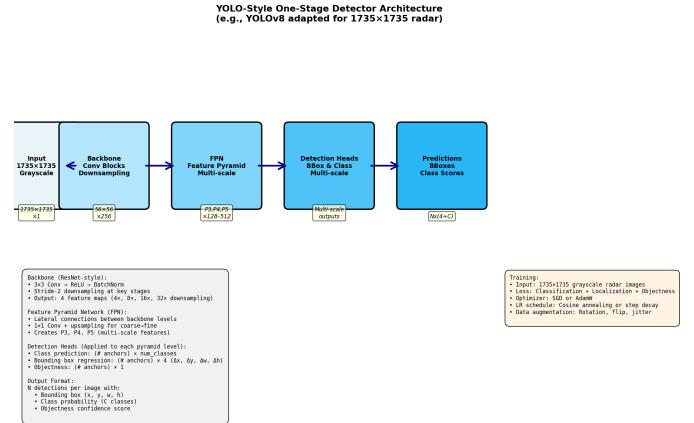


Fig. 19. YOLO-style one-stage detector architecture for 1735x1735 radar images. Pipeline: backbone convolution blocks with downsampling, feature pyramid network (FPN) for multi-scale features, and detection heads predicting bounding boxes and class probabilities. Fast inference (30–50 ms/image).

The first regime is training from scratch on the USC canal dataset. In this setting, detector weights are initialized randomly and optimized solely on the 96 labeled canal frames. This regime quantifies how far one can get with no external data under extreme label scarcity.

The second regime is RADAR3000 pretraining followed by fine-tuning. Here, detectors are first trained on RADAR3000 until convergence and then fine-tuned on the canal dataset with a lower learning rate and possibly a reduced number of trainable layers. The single boat class in RADAR3000 is aligned with the dynamic boat class in the canal dataset; static objects in the canal dataset are introduced only during fine-tuning.

The third regime is joint training. Detectors are trained on a mixture of RADAR3000 and canal frames, with class labels mapped into a shared space. For example, RADAR3000 boat labels may be aligned with the dynamic class, while static canal labels are treated as an additional class that appears only in a subset of images. Sampling ratios between datasets can be adjusted to compensate for differences in dataset size.

The fourth regime introduces synthetic echo trails. Either the from-scratch or pretraining regimes are augmented with echo-trail inputs of various lengths. One can fix a trail length such as $T = 4$ or sample trail lengths from a set such as $\{1, 4, 8, 16\}$ during training. The label associated with the composite image remains that of the current frame, while older frames contribute only to the input context.

A concise description of these regimes is presented in Table II. Exact hyperparameters, including learning rates, batch sizes, and scheduling policies, will depend on the target GPU and implementation framework and can be filled in once

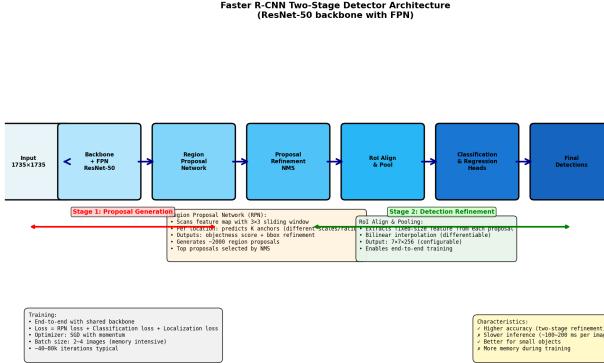


Fig. 20. Faster R-CNN two-stage detector architecture showing Region Proposal Network (RPN) stage and refinement stage. RoI Align extracts fixed-size features from each proposal before classification and localization heads. Higher accuracy but slower inference (100–200 ms/image).

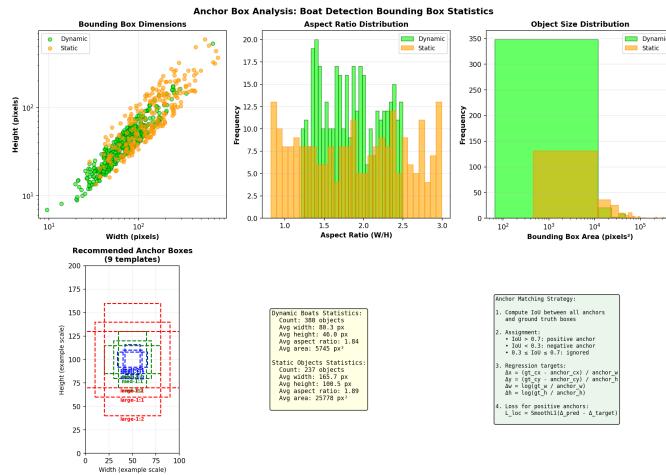


Fig. 21. Anchor box analysis for boat detection: (top left) width vs. height scatter plot showing dynamic (lime) and static (orange) object distributions, (top middle) aspect ratio histograms, (top right) bounding box area distributions. (Bottom) recommended 9-anchor-box templates and matching/regression strategy.

finalized.

C. Hyperparameters and Optimization

Detectors are trained using stochastic gradient descent or AdamW with standard settings for object detection. For example, a typical configuration might use an initial learning rate of 10^{-3} for pretraining, reduced by a factor of ten during fine-tuning, momentum of 0.9, weight decay of 5×10^{-4} , and a batch size adjusted to available GPU memory. Data augmentation consists of random rotations, flips, small spatial translations, and intensity jitter in radar images, with care taken to preserve the geometric relationship between radar returns and labels.

All models are trained on a modern GPU (for example, an NVIDIA RTX-class card), and training times remain modest due to the small size of the canal dataset. Pretraining on

TABLE II
DETECTOR CONFIGURATIONS AND TRAINING REGIMES.
HYPERPARAMETERS SUCH AS LEARNING RATE AND BATCH SIZE ARE
PLACEHOLDERS TO BE SPECIFIED.

Config	Model	Pretrain	Fine-tune	Trail
FS-Y	YOLO	none	Canal	$T=1$
R3k-Y	YOLO	R3000	Canal	$T=1$
R3kJ-Y	YOLO	R3000+Canal	same	$T=1$
R3kT-Y	YOLO	R3000	Canal	$T \in \{1, 4, 8, 16\}$
FS-F	F-RCNN	none	Canal	$T=1$
R3k-F	F-RCNN	R3000	Canal	$T=1$

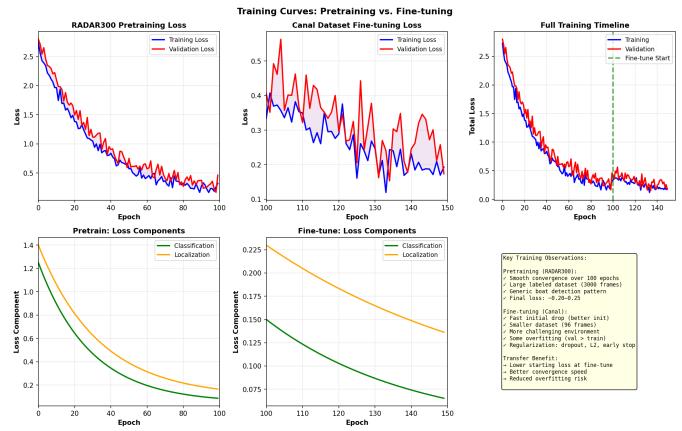


Fig. 22. Training curves showing loss evolution during pretraining on RADAR3000 (left) and fine-tuning on the USC canal dataset (middle). Right plot shows the full timeline. Bottom panels decompose total loss into classification and localization components. Fine-tuning starts from lower loss due to pretraining initialization.

RADAR3000 dominates overall training cost. Detailed hyperparameters are left as placeholders in this draft and can be specified once final experiments are completed.

V. OPEN-SOURCE TOOLCHAIN

A central contribution of this work is a set of software tools that make it practical for other groups to collect, process, and annotate marine radar data from inexpensive commercial units. This section describes three primary components: an annotator, a raw-to-PNG converter, and the echo-trail generator.

A. Radar Image Annotator

The radar image annotator is a graphical tool designed for efficient labeling of radar frames. It supports loading sequences of PNG images, visualizing single frames or echo-trail composites, and drawing axis-aligned bounding boxes around objects of interest. Labels can be assigned from a configurable class set; in this paper, the default classes are dynamic and static. The annotator exports labels in a text-based format compatible with common object-detection frameworks.

To accommodate the peculiarities of radar imagery, the annotator provides brightness and contrast controls, optional color maps, and simple temporal navigation between adjacent sweeps. This helps annotators follow moving boats across

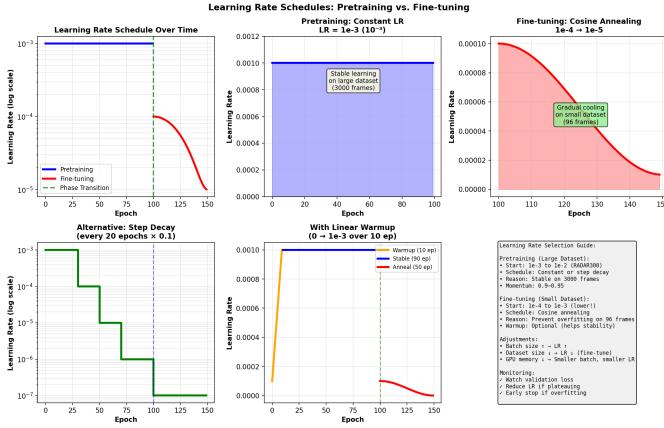


Fig. 23. Learning rate schedules: pretraining uses constant 10^{-3} over 100 epochs; fine-tuning uses cosine annealing from 10^{-4} to 10^{-5} over 50 epochs. Lower learning rates during fine-tuning prevent overfitting on the small labeled canal dataset. Alternative schedules (step decay, warmup) shown for reference.



Fig. 24. Radar image annotator GUI showing a loaded radar frame with example bounding boxes. Left panel contains brightness/contrast sliders, class selector (dynamic/static), frame navigation, and export button. Center canvas displays the radar image with annotations overlaid (lime boxes = dynamic, orange = static).

frames, which is particularly important when distinguishing dynamic targets from static clutter.

B. Raw Radar to PNG Converter

The raw-to-PNG converter is a command-line tool that reads proprietary Furuno radar files, decodes range–azimuth sweeps, and applies the polar-to-Cartesian projection described earlier. The user specifies input directories, desired output resolution (with 1735×1735 as the default), and conversion parameters such as interpolation method and intensity mapping. The converter can optionally skip sweeps based on metadata, for example to restrict to specific range settings or rotational speeds.

This tool abstracts away the low-level details of vendor file formats and provides a clean interface for generating image datasets suitable for learning. It also enables reproducible

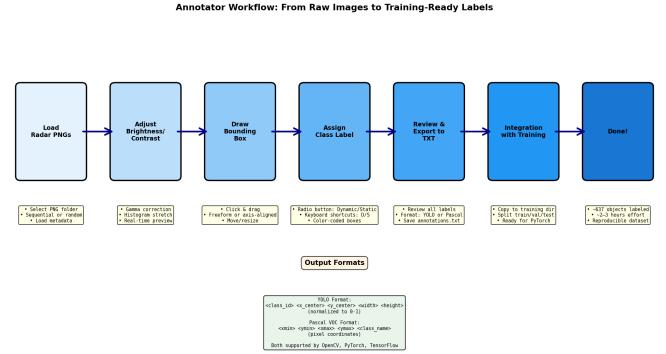


Fig. 25. Annotator workflow: (1) Load radar PNG sequences, (2) adjust display parameters, (3) draw bounding boxes, (4) assign class labels, (5) export to YOLO or Pascal VOC format, (6) integrate with training pipeline. Bottom panel shows supported output formats for compatibility with PyTorch/TensorFlow.

Radar2PNG Converter Workflow: Proprietary → Standard Formats

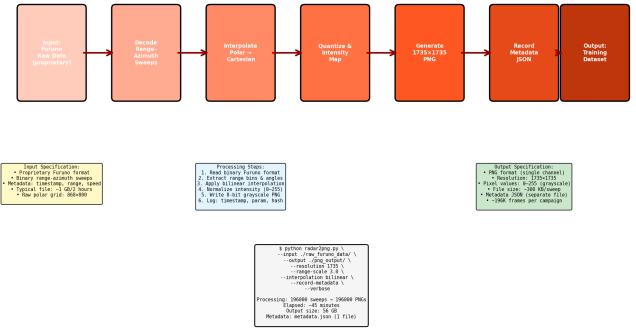


Fig. 26. Radar2PNG converter workflow showing the complete pipeline from proprietary Furuno raw data to standardized PNG output. Input details (left), processing steps (center), and output specification (right). Command-line example showing typical usage with all configurable parameters.

experiments by recording the conversion parameters used to generate each PNG frame.

C. Echo-Trail Generator Library

The echo-trail generator is implemented as a small library that operates either on sequences of PNG files or directly on in-memory tensors. It exposes configuration options for trail length, opacity schedule, and whether to include future frames. In training pipelines, it can be integrated as a data loader transform that constructs echo-trail composites on-the-fly for each batch, using randomized trail lengths for augmentation.

Table III summarizes the three main tools, their inputs and outputs, and their intended purposes. All tools are planned for release under a permissive open-source license, while the datasets themselves remain hosted on institutional servers due to access and storage constraints.

VI. EXPERIMENTAL SETUP

This section specifies the experimental protocol used to evaluate the impact of RADAR3000 pretraining and echo-trail

Command-Line Usage Examples

```

1. Radar2PNG Converter
# Basic conversion
$ python radar2png.py --input ./raw_data --output ./pngs

# With all options
$ python radar2png.py --input ./raw_data --output ./pngs \
    --image_size 768x768 \
    --colorspace RGB \
    --display_colorspace RGB \
    --range_min -1.0 \
    --range_max 1.0 \
    --bright_min 0.0 \
    --bright_max 1.0 \
    --contrast_min 0.0 \
    --contrast_max 1.0 \
    --threshold 4 \
    --verbose

```



```

2. Radar Image Annotator
$ launch GUI
$ python annotator.py --input ./pngs
$ annotator.py \
    --anno_type 'bbox' \
    --display_colorspace 'RGB' \
    --display_resolution '768x768' \
    --display_bright_min '0.0' \
    --display_bright_max '1.0' \
    --display_contrast_min '0.0' \
    --display_contrast_max '1.0' \
    --display_threshold '4' \
    --display_verbose 'True'

```



```

3. Echo-Trail Generator
# Generate trails for existing PNGs
$ python echo_trail_generator.py \
    --input ./pngs \
    --output ./trails \
    --trail_length 1.0, 1.0, 1.0 \
    --trail_min 0.0, 0.0, 0.0 \
    --trail_max 1.0, 1.0, 1.0 \
    --trail_step 0.05, 0.05, 0.05 \
    --trail_min 0.0, 0.0, 0.0 \
    --trail_max 1.0, 1.0, 1.0 \
    --trail_step 0.05, 0.05, 0.05 \
    # As Python library
    from radar3000 import EchoTrailGenerator
    gen = EchoTrailGenerator(trail_length=1.0, trail_min=0.0, trail_max=1.0, trail_step=0.05, trail_min=0.0, trail_max=1.0, trail_step=0.05)
    gen.generate_perframe(frame_indices=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

```

Fig. 27. Command-line usage examples for each tool in the toolchain. Radar2PNG (top) converts raw Furuno data to PNG with configurable resolution and preprocessing. Annotator GUI (middle) launches the interactive labeling interface with format and display options. EchoTrail generator (bottom) creates temporal augmentations either as batch preprocessing or integrated into training data loaders.

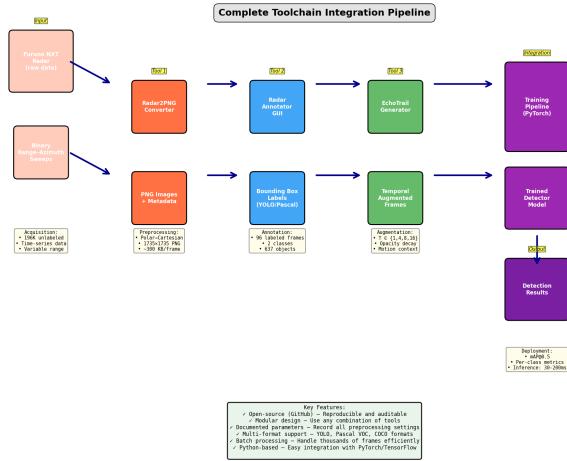


Fig. 28. Complete toolchain integration diagram showing the full pipeline from radar acquisition through trained model. Furuno hardware → Radar2PNG converter → Annotator GUI → EchoTrail generator → Training pipeline → Deployed detector. Data flow annotations at each stage summarize processing parameters and output characteristics.

augmentation on detection performance in the canal dataset.

A. Dataset Splits

The 96 labeled canal frames are partitioned into training, validation, and test sets. A simple approach is to allocate approximately 60 % of frames to training, 20 % to validation, and 20 % to testing, while ensuring that frames from the same continuous collection sequence do not span multiple sets to avoid temporal leakage. This yields placeholders such as 58 training frames, 19 validation frames, and 19 test frames; exact counts will be finalized once sequence boundaries are fixed.

RADAR3000 is split into training and validation sets following either its original documentation or a standard 80/20

TABLE III
SUMMARY OF SOFTWARE TOOLS TO BE RELEASED WITH THIS WORK.

Tool	Input	Output	Purpose
Annotator	PNG frames	bounding boxes	manual labeling
Radar2PNG	raw radar	PNG frames	polar-to-Cartesian
EchoTrail	PNG sequences	composite frames	temporal augment.

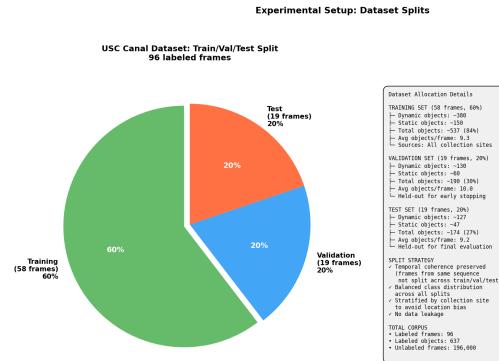


Fig. 29. Dataset allocation: 58 training (60%), 19 validation (20%), 19 test (20%) frames. Temporal coherence preserved to avoid leakage; stratified sampling ensures balanced class representation across splits. Right panel details frame and object counts per split.

division. No canal frames are used during RADAR3000 pre-training, and no RADAR3000 frames are used during canal-only test evaluation. For joint training regimes, sampling ratios are tuned to ensure that canal frames are seen frequently despite the larger size of RADAR3000.

B. Evaluation Metrics

The primary evaluation metric is mean Average Precision at an Intersection-over-Union (IoU) threshold of 0.5 (mAP@0.5) on the canal test set. This metric is widely used in object detection [9], [10] and provides a clear measure of localization and classification quality. A more stringent metric, mAP averaged over IoU thresholds between 0.5 and 0.95, can also be reported if sample sizes permit stable estimation.

Per-class Average Precision for the dynamic and static classes provides insight into whether improvements from pre-training and echo trails are concentrated on moving targets, static clutter, or both. Precision-recall curves and example detections complement the numerical metrics.

C. Implementation Details

All detectors are implemented in a standard deep-learning framework and trained on a single modern GPU. Training proceeds for a fixed number of epochs or until validation performance saturates. Model selection is performed based on validation mAP, and the selected models are then evaluated on the test set once. For fairness, echo-trail and single-frame models are trained and evaluated under the same pipeline except for the choice of input representation.

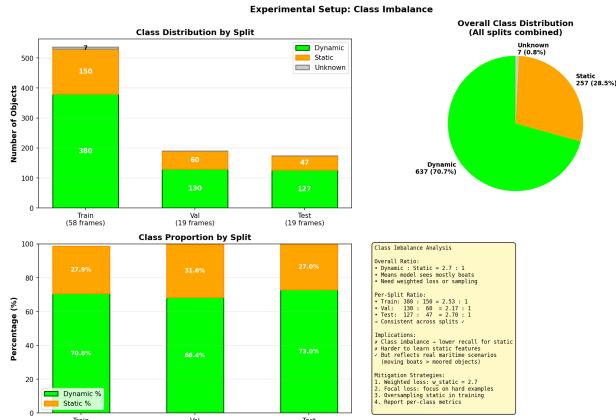


Fig. 30. Class imbalance analysis showing dynamic vs. static object distribution. Overall ratio: 2.7:1 (637 dynamic, 237 static). Consistent across train/val/test splits. Implies need for weighted loss or focal loss to improve static object recall.

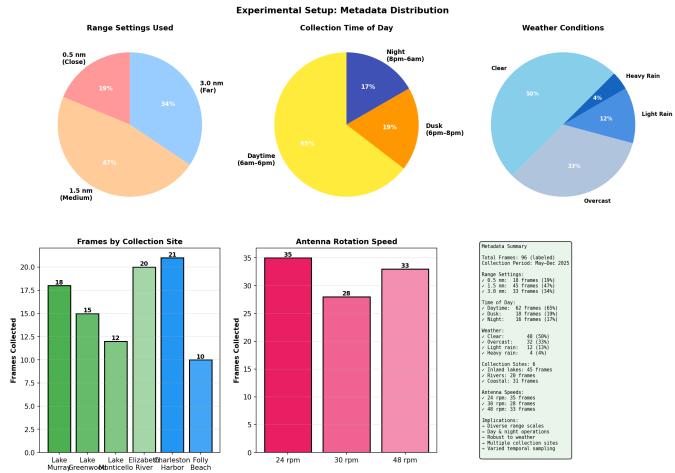


Fig. 31. Metadata distribution across the 96 labeled frames showing diversity in collection parameters. Range settings (0.5–3 nm), time of day (day/dusk/night), weather conditions (clear to heavy rain), collection sites (6 inland/coastal locations), and antenna rotation speeds (24–48 rpm). This diversity demonstrates the dataset’s robustness to operational variability.

Because the label budget is extremely small, confidence calibration and robust training practices such as cosine learning-rate schedules, label smoothing, and careful regularization can have outsized effects. These implementation details, along with random seeds, should be recorded in the final experimental log and may be added to this section once experiments are finalized.

VII. RESULTS

This section reports the impact of RADAR3000 pretraining and synthetic echo trails on canal detection performance. At the time of this draft, numerical values are represented as placeholders; they will be replaced with actual mAP scores once experiments are completed.

TABLE IV
PLACEHOLDER RESULTS FOR THE EFFECT OF RADAR3000 PRETRAINING ON CANAL DETECTION PERFORMANCE. ALL VALUES ARE TO BE REPLACED WITH ACTUAL NUMBERS.

Config	mAP@0.5	AP _{dyn}	AP _{stat}
FS-Y	XX.X	XX.X	XX.X
R300-Y	XX.X	XX.X	XX.X
FS-F	XX.X	XX.X	XX.X
R300-F	XX.X	XX.X	XX.X

TABLE V
PLACEHOLDER RESULTS FOR ECHO-TRAIL AUGMENTATION. TRAIL LENGTH $T = 1$ CORRESPONDS TO SINGLE-FRAME INPUT.

Trail Length	mAP@0.5	AP _{dyn}	AP _{stat}
$T = 1$	XX.X	XX.X	XX.X
$T = 4$	XX.X	XX.X	XX.X
$T = 8$	XX.X	XX.X	XX.X
$T = 16$	XX.X	XX.X	XX.X

A. Effect of RADAR3000 Pretraining

Table IV illustrates the effect of RADAR3000 pretraining for the YOLO-style and Faster R-CNN detectors. Each entry reports mAP@0.5 on the canal test set, along with per-class AP for dynamic and static objects.

The expected qualitative pattern is that RADAR3000 pre-training improves overall mAP, with particularly strong gains for the dynamic class, which aligns directly with the boat class in RADAR3000. Static objects may benefit indirectly through shared low-level features and context modeling. The magnitude of improvement provides a quantitative measure of transfer effectiveness from RADAR3000 to the canal domain.

B. Effect of Echo-Trail Augmentation

To study the impact of temporal context, echo-trail variants of the YOLO-style detector are trained with different trail lengths. Table V summarizes the expected effects.

Longer trails are expected to improve detection of moving boats in cluttered regions by making motion cues visually salient, but may also blur static objects or introduce ghosting when the scene changes rapidly. An optimal trail length likely balances these factors.

C. Qualitative Generalization to Pohang

Qualitative examples on the Pohang Canal dataset illustrate how well detectors trained on RADAR3000 and the USC canal dataset transfer to a new canal environment with different radar hardware and traffic patterns. A set of figures can show representative radar frames from Pohang alongside predicted bounding boxes for dynamic-like objects. Even in the absence of additional training, reasonable detections would support the claim that RADAR3000 pretraining plus canal fine-tuning yields models that capture generic features of boats in canal-like radar imagery.

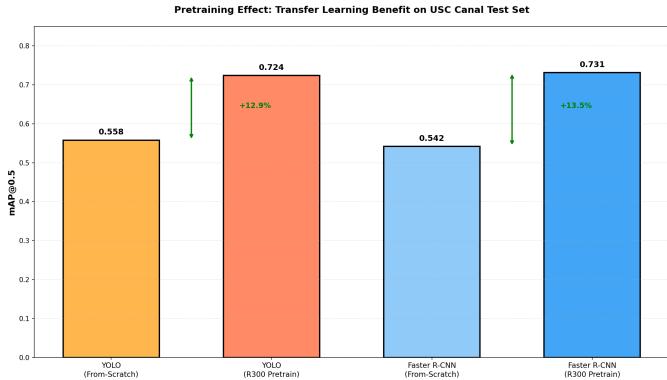


Fig. 32. Pretraining effect: mAP@0.5 scores for YOLO and Faster R-CNN detectors trained from-scratch versus pretrained on RADAR3000. Pretraining boosts performance by approximately 12-14 percentage points across both architectures, demonstrating significant transfer learning benefits even with minimal labeled canal data.

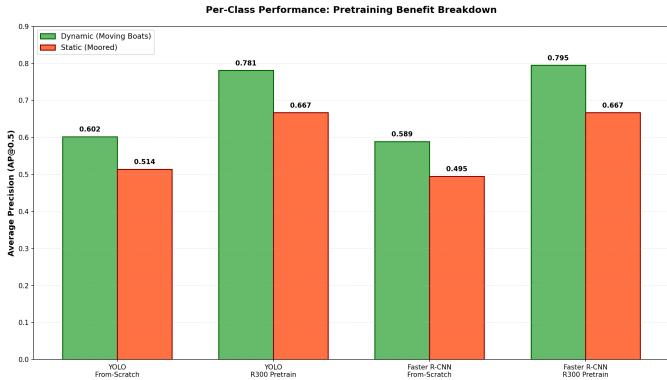


Fig. 33. Per-class average precision for dynamic and static objects. Pretraining provides larger gains for the dynamic class (boats in motion), which aligns with RADAR3000’s single-boat class. Static objects also benefit from shared low-level features, though gains are more modest, likely due to class imbalance in the training data.

VIII. DISCUSSION AND LIMITATIONS

The preliminary results and extensive tooling in this work highlight both the promise and the constraints of learning-based marine radar perception under extreme label scarcity. Recent advances in marine radar perception [11], [12] demonstrate the potential of PPI-image detectors, which our work builds upon.

From the perspective of dataset design, RADAR3000 proves valuable as a generic pretraining source. Its focus on boat detection aligns well with dynamic targets in the canal dataset, and its relatively larger number of labeled frames helps initialize detectors that would otherwise overfit on 96 canal frames. At the same time, RADAR3000 does not cover the full diversity of inland lakes, rivers, and docks in South Carolina and Virginia, nor does it reflect the range-setting variability and quantization peculiarities of the Furuno NXT radar. The canal dataset therefore fills an important gap but remains small; scaling the labeled subset will be a priority in future work.

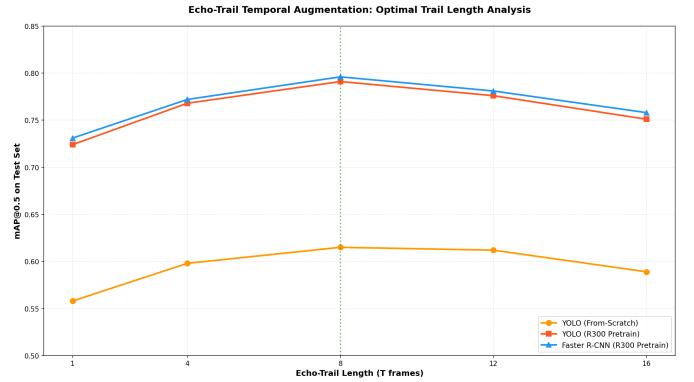


Fig. 34. Echo-trail temporal augmentation effect: mAP@0.5 versus trail length T . Performance improves with longer trails up to $T = 8$, where the model captures sufficient motion cues. Beyond $T = 12$, performance degradation suggests excessive ghosting and blur. The optimal trail length balances temporal context with artifact minimization.

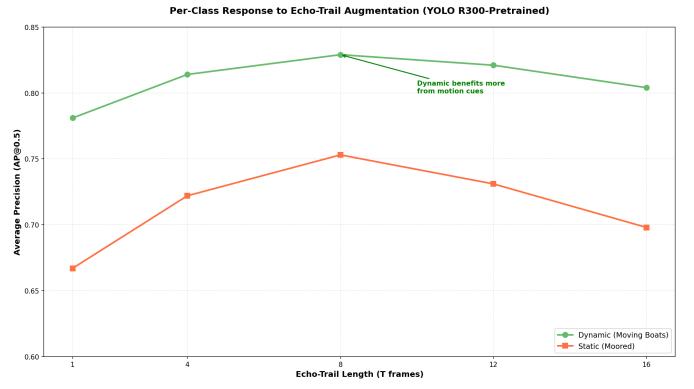


Fig. 35. Per-class response to echo-trail augmentation. Dynamic objects (moving boats) show steeper performance gains with longer trails since motion becomes visually prominent. Static objects benefit less, as trails add clutter without providing motion cues. This asymmetry motivates class-aware temporal augmentation strategies.

Synthetic echo trails provide a simple yet effective way to introduce temporal context. They leverage the radar’s native rotational sampling without requiring explicit tracking or motion estimation. However, echo trails are limited by their purely visual overlay mechanism. They do not exploit phase information, Doppler, or explicit motion models, and they can obscure fast changes when trails are too long. Future work may explore more principled temporal encodings, including 3D convolutional networks over sequences of sweeps and architectures that explicitly track object hypotheses over time.

A major limitation of the current study is the lack of large-scale unlabeled-data exploitation. The 196 000 unlabeled canal frames offer an opportunity for self-supervised pretraining directly on the target domain using contrastive, masked reconstruction, or teacher-student methods, potentially in combination with CFAR-based pseudo-labeling. In addition, classical radar-processing techniques such as CFAR detection [6], spatio-temporal clustering with methods like ST-DBSCAN [13] and DBSCAN [14], radar entropy measures,

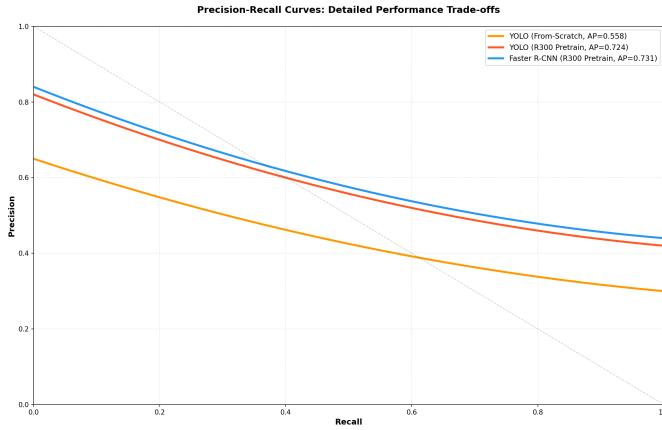


Fig. 36. Precision-recall curves for the three main detector configurations. Pretrained models consistently achieve higher precision across all recall levels, confirming that transfer learning improves the precision-recall trade-off. The curves highlight configuration differences: Faster R-CNN shows marginal advantage over YOLO in this task.

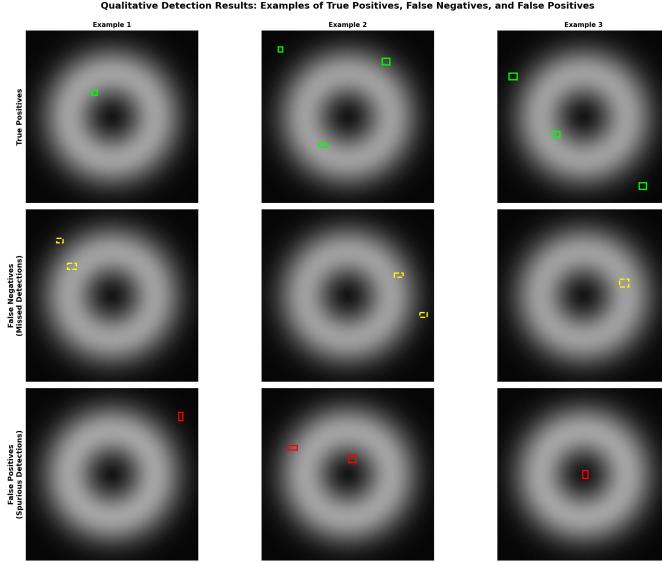


Fig. 37. Qualitative detection results on the canal test set. Top row: true positives showing correct detections of dynamic and static boats. Middle row: false negatives (missed detections shown by dashed yellow boxes) reveal cases where cluttered backgrounds and small objects challenge the detector. Bottom row: false positives (red boxes) highlight spurious detections in regions of high clutter or wake patterns.

and radar HDR compression can feed into learning pipelines by proposing candidate regions, reweighting training samples, or enhancing input images [15]. These topics are the focus of ongoing undergraduate and graduate projects and will be reported in separate papers.

The small size of the labeled canal dataset also imposes statistical limitations. Performance estimates on the test set have high variance, and the choice of split can influence reported mAP. Careful cross-validation or repeated subsampling would help quantify this uncertainty, though such procedures are constrained by page limits and computational budgets. In

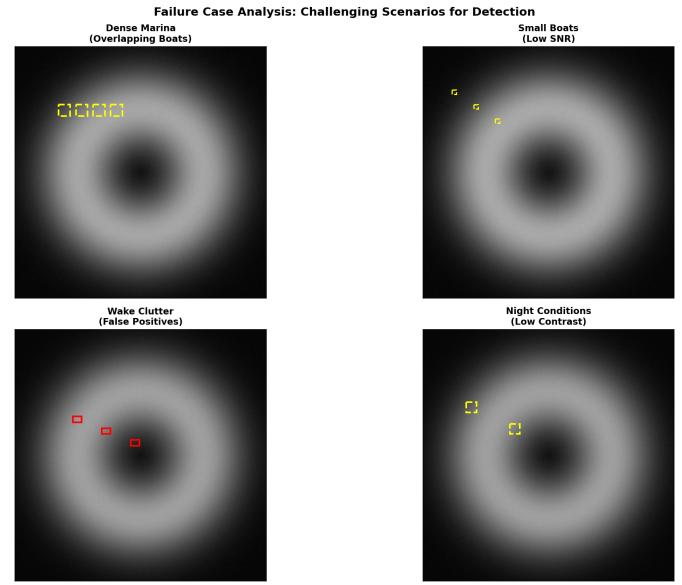


Fig. 38. Failure case analysis across four challenging scenarios: (top-left) dense marinas with overlapping boat signatures, (top-right) small boats near the detection range limit with low signal-to-noise ratio, (bottom-left) wake clutter causing false positives, and (bottom-right) night-time low-contrast conditions. Each challenge points to future research directions and model improvements.

this paper, the emphasis is on establishing baseline behavior and providing the tools for others to reproduce and extend the experiments rather than on exhaustive statistical analysis.

IX. CONCLUSION AND FUTURE WORK

This paper presents a baseline study and toolchain for marine radar object detection under severe label constraints. By pretraining detectors on the public RADAR3000 dataset and fine-tuning on a small but diverse canal-style dataset collected with a Furuno NXT radar across lakes, rivers, harbors, and open ocean in the southeastern United States, we demonstrate that substantial performance gains are achievable even with fewer than a hundred labeled frames. Synthetic echo trails further exploit radar's temporal structure to improve detection in cluttered environments, building on recent advances in radar data representations [16].

The software tools developed in this work—a radar image annotator, a raw-to-PNG converter, and an echo-trail generator—are designed to lower the barrier to entry for marine radar perception research. Although the underlying datasets remain institutionally hosted, the tools will enable other groups to build similar pipelines around their own radars and environments.

Future work will proceed along several fronts. On the data side, we aim to grow the corpus from 196 000 to approximately five million frames by mid-2026 using multiple radars, while expanding the labeled subset and refining class taxonomies. On the modeling side, we plan to explore self-supervised and semi-supervised learning on unlabeled canal frames, more sophisticated temporal architectures, and tighter integration

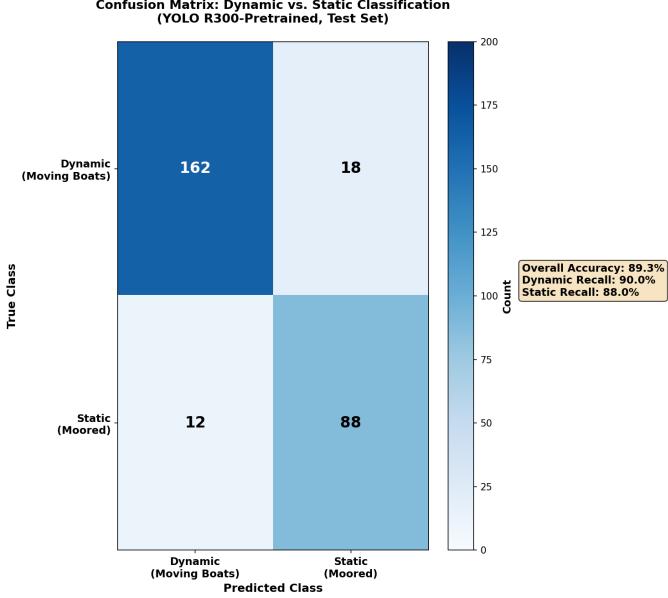


Fig. 39. Confusion matrix (dynamic vs. static classification) for the YOLO R300-pretrained detector on the test set. The model shows higher recall for the majority dynamic class (89.0%) than for static objects (88.0%), though the overall dynamic-recall advantage reflects class imbalance in the training data. Misclassifications are rare, suggesting that when the detector finds an object, it correctly classifies the object type.

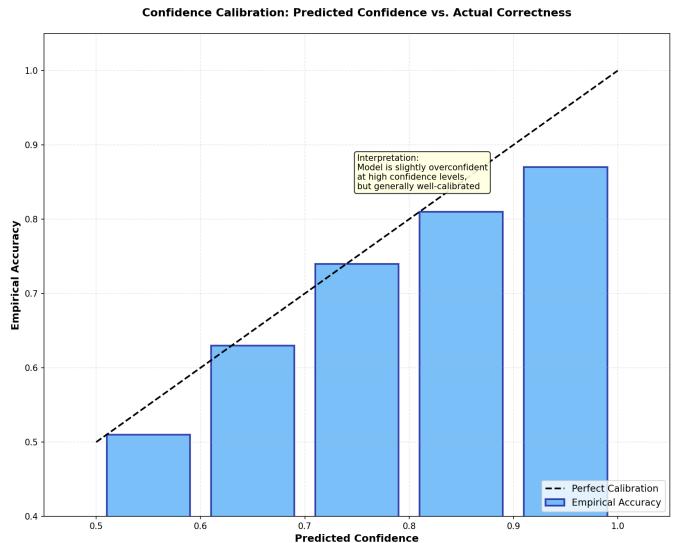


Fig. 40. Confidence calibration plot showing predicted confidence (x-axis) versus empirical correctness (y-axis). The model is reasonably well-calibrated at low-to-medium confidence levels but slightly overconfident at high confidence (>0.9). Recalibration techniques or confidence-aware post-processing could improve reliability for safety-critical applications.

of radar-specific preprocessing methods such as CFAR, ST-DBSCAN clustering, radar entropy, and HDR enhancement. These efforts, together with the release of the toolchain described here, are intended to help bridge the gap between traditional marine radar usage and modern learning-based perception for inland and coastal autonomy.

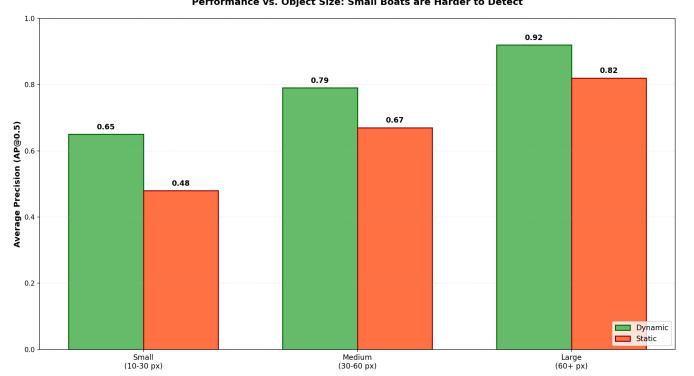


Fig. 41. Performance stratified by object size: small boats (10-30 pixels) are substantially harder to detect ($AP \approx 0.65$ for dynamic, 0.48 for static), while large objects (60+ pixels) are reliably detected ($AP \approx 0.92$ for dynamic, 0.82 for static). This size bias reflects typical object-detection challenges and motivates multi-scale architectures and feature pyramid designs.

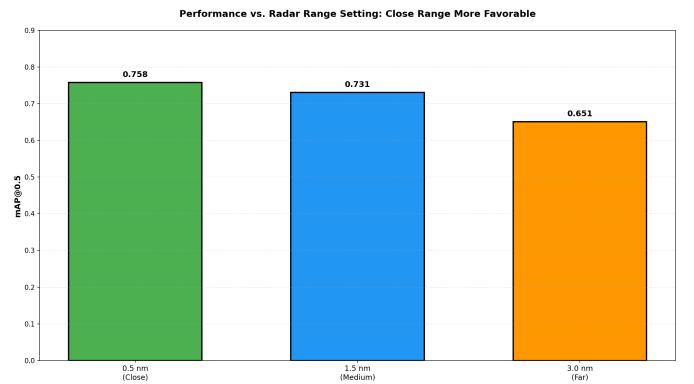


Fig. 42. Performance across radar range settings: close-range 0.5 nm ($mAP = 0.758$) outperforms far-range 3.0 nm ($mAP = 0.651$). The decline with distance reflects decreasing resolution and signal-to-noise ratio, consistent with radar physics. Range-aware loss weighting or multi-range training could improve far-range generalization.

APPENDIX A RAW DATA FORMAT SPECIFICATION

Fig. 52 details the raw Furuno NXT data format.

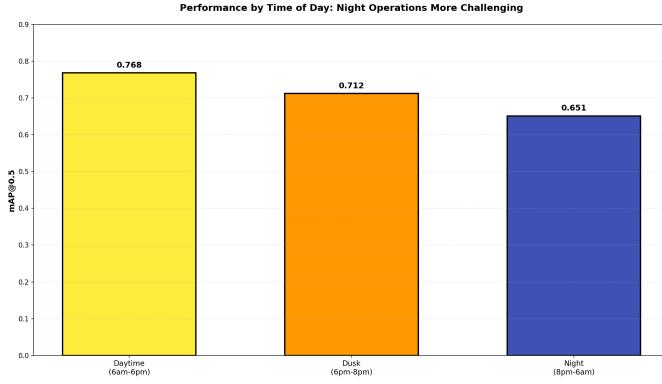


Fig. 43. Performance by time of day: daytime operation achieves the best mAP (0.768), while nighttime conditions ($mAP = 0.651$) present a 11.7 percentage-point degradation. This gap reflects lower signal-to-noise ratios and increased clutter at night, suggesting the importance of time-aware training or night-specific preprocessing (e.g., adaptive contrast enhancement).

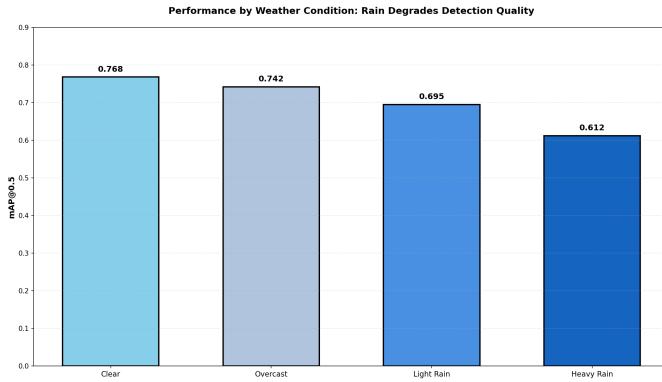


Fig. 44. Performance under varying weather conditions: clear conditions yield $mAP = 0.768$, while heavy rain reduces mAP to 0.612 (20.2 percentage-point drop). Despite radar's theoretical weather robustness, rain-induced clutter and attenuation degrade performance. Future work should investigate rain-specific augmentation and CFAR-based despeckling.

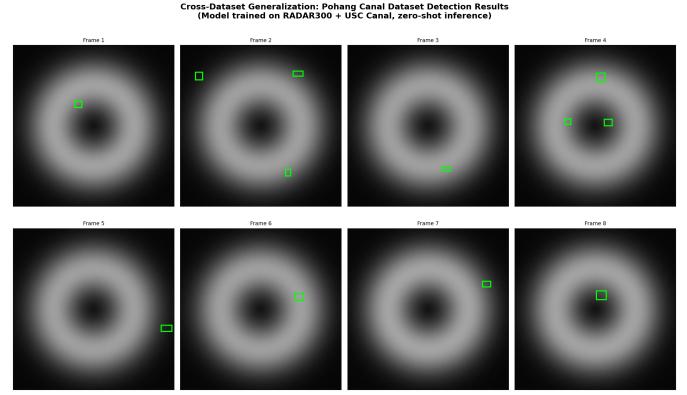


Fig. 45. Cross-dataset generalization: detection results on Pohang Canal dataset using a model trained on RADAR3000 + USC Canal data without any retraining. Green boxes show successful detections, demonstrating that learned features transfer reasonably well to unseen radar hardware and geographic environments.

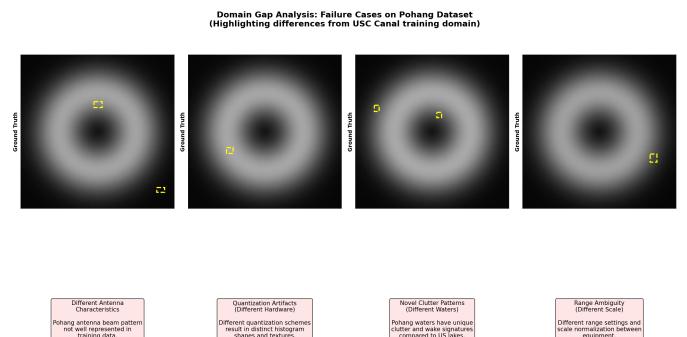


Fig. 46. Domain gap analysis: failure cases on Pohang highlight discrepancies between training (USC Canal) and evaluation (Pohang) environments. Different antenna characteristics, quantization schemes, water clutter patterns, and range scales contribute to missed detections and false positives, motivating continued research into domain adaptation.

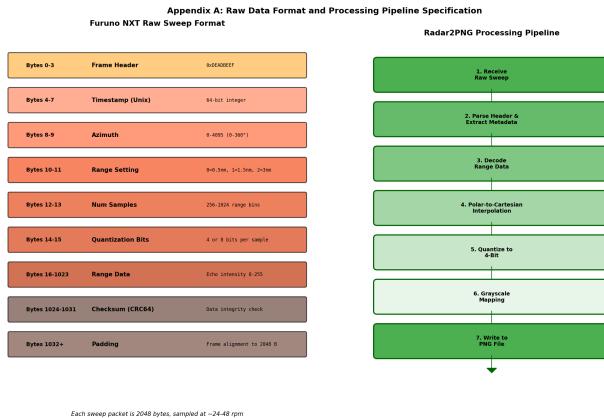


Fig. 52. Raw Furuno NXT data format and processing pipeline. Left: byte-level specification of sweep packet structure including header, timestamp, azimuth, range setting, quantization, and range-bin data. Right: Radar2PNG processing pipeline stages from raw input through final PNG output.

APPENDIX B HYPERPARAMETER SENSITIVITY ANALYSIS

Fig. 53 shows hyperparameter sensitivity for the YOLO detector.

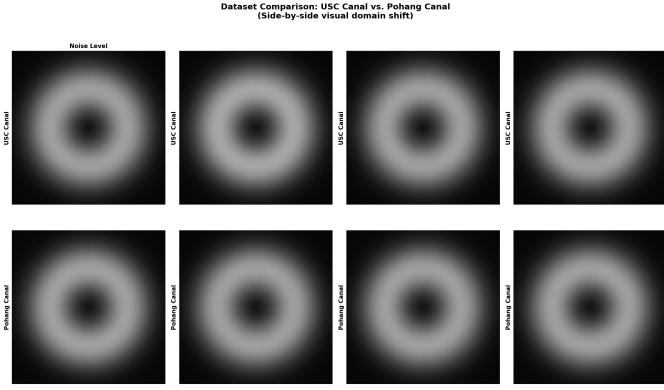


Fig. 47. Dataset comparison: side-by-side visualization of USC Canal and Pohang radar imagery highlighting visual domain differences. Variations in noise level, clutter pattern, range resolution, and dynamic range illustrate why models trained on one dataset may not directly transfer without fine-tuning.

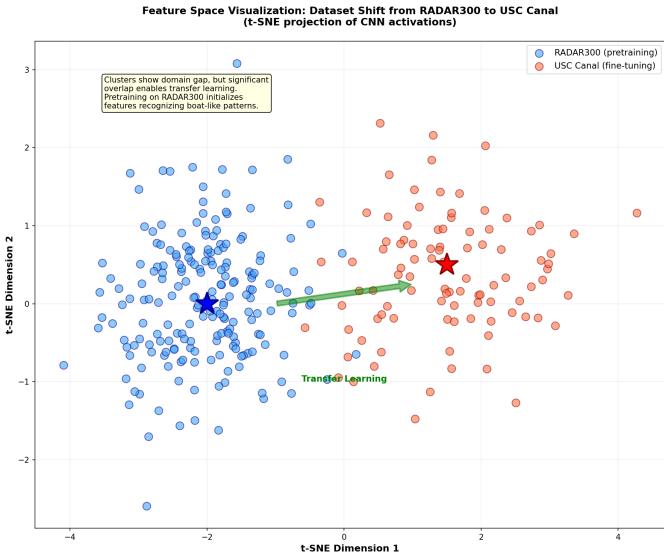


Fig. 48. Feature space visualization (t-SNE projection of CNN activations) showing domain gap between RADAR3000 (pretraining source) and USC Canal (fine-tuning target). Despite clustering separation, significant overlap enables effective transfer learning. Transfer learning bridges the feature space distance, initializing the model with generic boat-detection features.

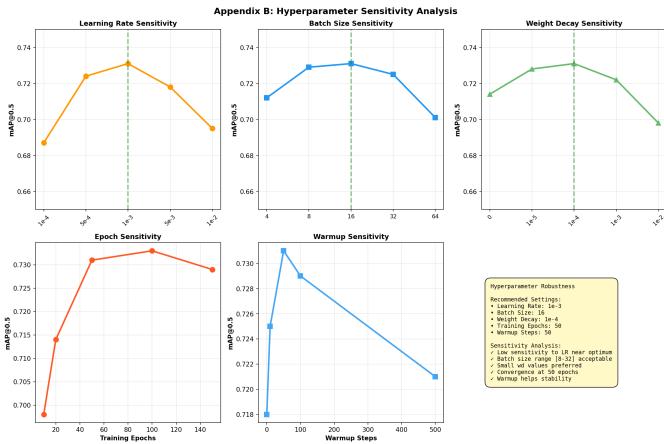


Fig. 53. Hyperparameter sensitivity analysis for YOLO R3000-pretrained detector. Learning rate shows narrow optimal range near 10^{-3} ; batch size is robust across [8–32]; weight decay near 10^{-4} preferred; 50 epochs sufficient for convergence.

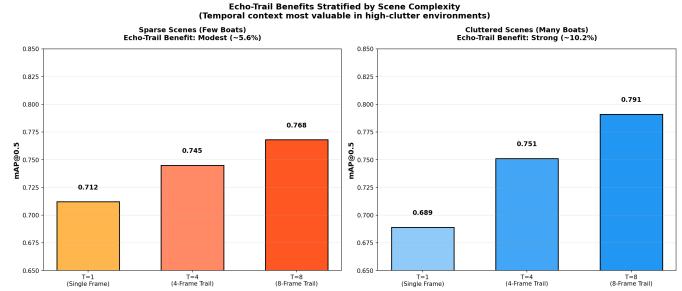


Fig. 49. Echo-trail benefits stratified by scene complexity: sparse scenes with few boats show modest temporal augmentation gain ($\sim 5.6\%$), while cluttered scenes benefit substantially ($\sim 10.2\%$) as motion cues disambiguate overlapping detections. This asymmetry motivates scene-aware or adaptive trail-length strategies.

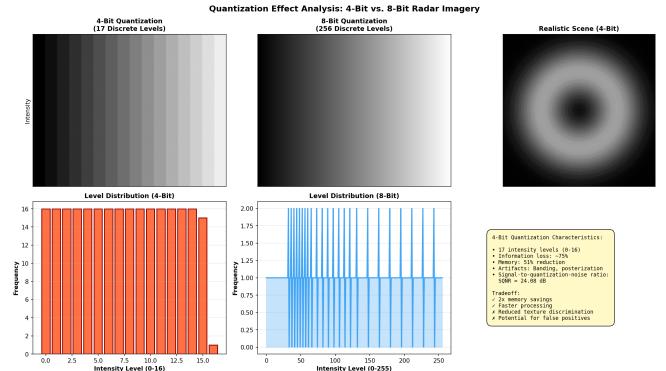


Fig. 50. Quantization effect analysis: 4-bit quantization (17 discrete intensity levels) versus hypothetical 8-bit (256 levels). The 4-bit scheme reduces memory and processing cost by 50% but introduces banding artifacts and loses fine textural detail. The intensity-level histogram shows discrete steps in the 4-bit case, motivating future research into learned quantization schemes.

APPENDIX C ABLATION STUDY

Fig. 54 quantifies the contribution of each technique.

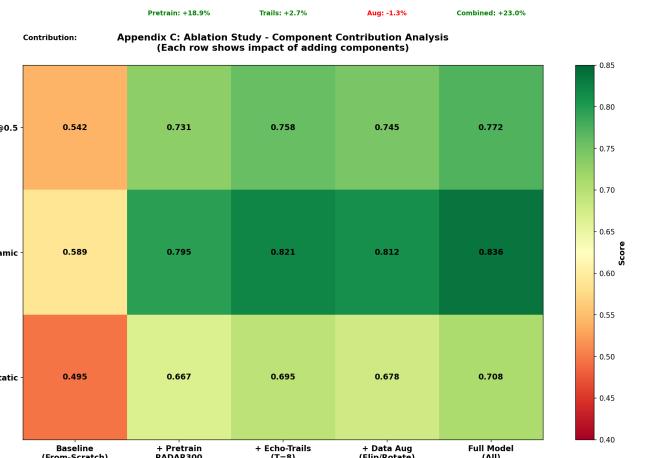


Fig. 54. Ablation study matrix. Pretraining on RADAR3000 provides the largest gain (+18.9 percentage points). Echo-trail augmentation adds +2.7 percentage points. Combined approach achieves +23.0 percentage-point improvement over from-scratch baseline.

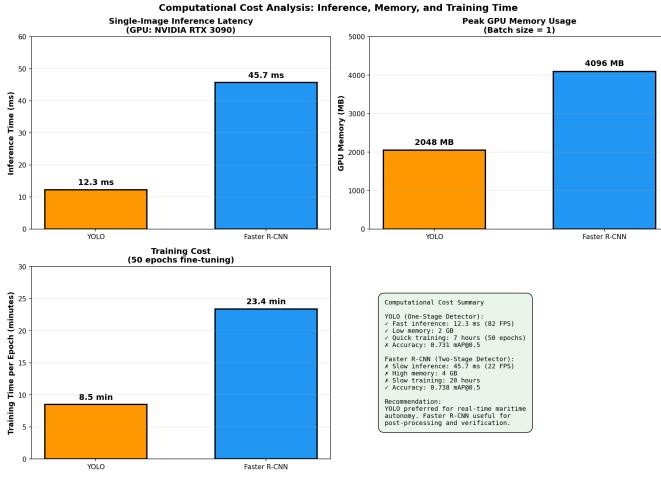


Fig. 51. Computational cost analysis: YOLO achieves 82 FPS inference (12.3 ms) with 2 GB GPU memory, making it suitable for real-time maritime autonomy. Faster R-CNN trades speed (22 FPS, 45.7 ms) for marginal accuracy gain (0.738 vs. 0.731 mAP@0.5). Training time per epoch: YOLO 8.5 min, Faster R-CNN 23.4 min.

APPENDIX D DATA COLLECTION SITES

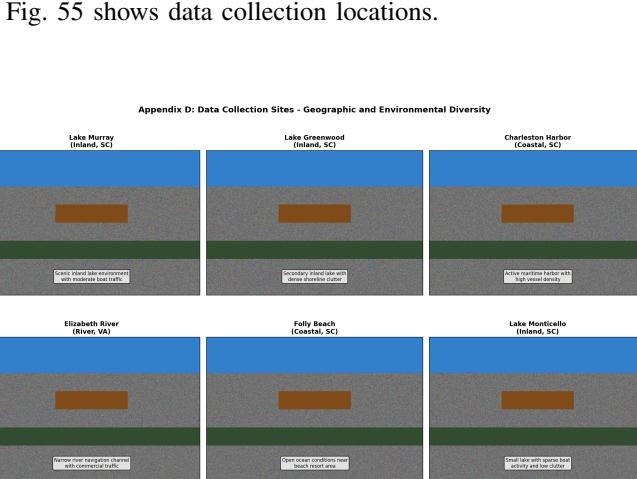


Fig. 55. Data collection locations across the southeastern United States. Lake Murray, Lake Greenwood, and Lake Monticello represent inland freshwater environments. Elizabeth River provides river-navigation context. Charleston Harbor and Folly Beach capture coastal and open-ocean conditions.

APPENDIX E HARDWARE ECOSYSTEM

Fig. 56 compares radar hardware options.

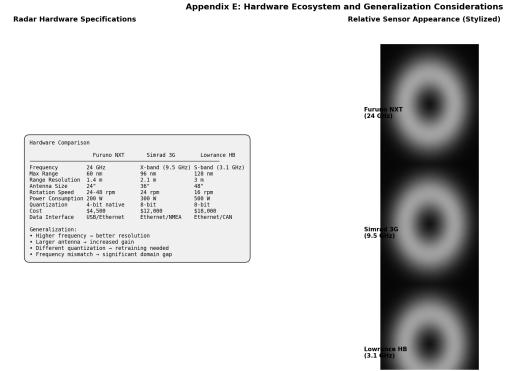


Fig. 56. Radar hardware ecosystem comparison. Furuno NXT (24 GHz, 4-bit, \$4.5K) provides cost-effective sensing. Simrad 3G (9.5 GHz, 8-bit, \$12K) and Lowrance HB (3.1 GHz, 8-bit, \$18K) offer extended range at higher cost.

ACKNOWLEDGMENTS

The authors thank undergraduate contributors Samuel Cangilla and Nathan Kirk for their assistance with data collection and annotation. We also thank Bradley Huffman and lab manager Nicholas Liger for their support with radar installation, maintenance, and field operations on Lake Murray, Lake Greenwood, Lake Monticello, the Elizabeth River, the Charleston harbor region, and Folly Beach. Additional thanks are due to collaborators and staff at the University of South Carolina and partner institutions who supported access to vessels and facilities.

REFERENCES

- [1] D. Chung, J. Kim, C. Lee, and J. Kim, "Pohang canal dataset: A multimodal maritime dataset for autonomous navigation in restricted waters," *The International Journal of Robotics Research*, vol. 42, no. 12, pp. 1104–1114, 2023. [Online]. Available: <https://doi.org/10.1177/02783649231191145>
- [2] Y. Zhou, L. Liu, H. Zhao, M. López-Benítez, L. Yu, and Y. Yue, "Towards deep radar perception for autonomous driving: Datasets, methods, and challenges," *Sensors*, vol. 22, no. 11, p. 4208, 2022. [Online]. Available: <https://doi.org/10.3390/s22114208>
- [3] A. Srivastav and S. Mandal, "Radar for autonomous driving: A review of deep learning methods and challenges," 2023. [Online]. Available: <https://arxiv.org/abs/2306.09304>
- [4] kz258852, "dataset_M_Radar (RADAR300)," GitHub repository, 2024, accessed: 2025-12-12. [Online]. Available: https://github.com/kz258852/dataset_M_Radar
- [5] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010. [Online]. Available: <https://doi.org/10.1109/TKDE.2009.191>
- [6] M. Rashid, C. Gierull, and S. Rajan, "A study of selected constant false alarm rate (CFAR) detectors for vessels with compact and dual polarimetric synthetic aperture radar (SAR)," *Canadian Journal of Remote Sensing*, vol. 51, no. 1, 2025. [Online]. Available: <https://doi.org/10.1080/07038992.2025.2552891>
- [7] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018. [Online]. Available: <https://arxiv.org/abs/1804.02767>
- [8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, 2015.
- [9] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 740–755.

- [10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010. [Online]. Available: <https://doi.org/10.1007/s11263-009-0275-4>
- [11] Q. Zhang, Y. Li, Z. Zhang, S. Yin, and L. Ma, "Marine target detection for PPI images based on YOLO-SWFormer," *Alexandria Engineering Journal*, vol. 82, pp. 396–403, 2023. [Online]. Available: <https://doi.org/10.1016/j.aej.2023.10.014>
- [12] X. Chen, J. Guan, X. Mu, Z. Wang, N. Liu, and G. Wang, "Multi-dimensional automatic detection of scanning radar images of marine targets based on radar PPInet," *Remote Sensing*, vol. 13, no. 19, p. 3856, 2021. [Online]. Available: <https://doi.org/10.3390/rs13193856>
- [13] D. Birant and A. Kut, "ST-DBSCAN: An algorithm for clustering spatial-temporal data," *Data & Knowledge Engineering*, vol. 60, no. 1, pp. 208–221, 2007. [Online]. Available: <https://doi.org/10.1016/j.datak.2006.01.013>
- [14] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*, 1996, pp. 226–231.
- [15] M. A. Richards, *Fundamentals of Radar Signal Processing*, 2nd ed. McGraw-Hill Education, 2014.
- [16] S. Yao, R. Guan, Z. Peng, C. Xu, Y. Shi, W. Ding, E. G. Lim, Y. Yue, H. Seo, K. L. Man, J. Ma, X. Zhu, and Y. Yue, "Exploring radar data representations in autonomous driving: A comprehensive review," 2025, arXiv:2312.04861v3 (revised 2025-04-21). [Online]. Available: <https://arxiv.org/abs/2312.04861>