RESEARCH ARTICLE · IEEE ACCESS, VOLUME 14 · 2026

Enhancing Radar PPI-Based Maritime Object Detection Using Hybrid Preprocessing and Deep Learning Approach

Helping YOLO models see small, low-contrast buoys through sea clutter

Sercan Külcü
Department of Computer Engineering, Giresun University, Türkiye

IEEE Access, vol. 14, pp. 111715–111730, 2026 · doi:10.1109/ACCESS.2026.3715472

SUMMARY

What Did This Study Achieve?

1 The Method

A hybrid pipeline: domain-specific image preprocessing combined with YOLOv8s and YOLOv12s, enhanced with a P2 feature layer and an area-aware loss.

2 Simulated-Data Result

The best configuration reaches 0.958 mAP50 and 0.812 mAP50-95 on 350 simulated radar frames.

3 Real-Data Result

On the real-world DAAN dataset, the same ideas push mAP50 from 0.698 (raw baseline) up to 0.922.

4 Why It Matters

The biggest gains are for small, low-contrast buoys — exactly the objects that are easiest to miss and most critical for safe navigation.

Why Do Standard Detectors Struggle on Radar?

Radar ≠ Camera Images

Radar PPI images have diffuse boundaries, grey-scale intensity instead of colour, and clutter patterns that look like real targets.

Small, Weak Targets

Buoys are small and low-contrast. Off-the-shelf YOLO models tend to miss them or place loose, inaccurate boxes.

The gap: classic image-processing fixes (thresholding, morphology) exist, but they are rarely built into an end-to-end deep learning pipeline — and newer YOLO variants are rarely tested specifically on radar PPI data.

This study connects the two worlds: it builds domain-specific preprocessing directly into a modern YOLO pipeline, and adds architecture changes aimed squarely at small, low-contrast maritime targets.

Source: Section I (Külcü 2026)

CONTRIBUTION

Four Contributions to Maritime Radar Detection

A high-resolution P2 layer

Added to selected YOLO models to keep fine spatial detail for small and low-contrast objects.

A task-specific, area-aware loss

Gives extra training weight to bounding boxes smaller than 8×8 pixels — exactly the safety-critical small targets.

A five-model comparison

YOLOv5s, v8s, v10s, v11s, and v12s, all evaluated on the same 350-frame simulated radar dataset.

Real-world external validation

The same pipeline is tested on the real DAAN marine radar dataset — not just simulated data.

A 350-Frame Simulated Radar Dataset

Built with a real bridge simulator, not hand-drawn images

- Frames come from the Wärtsilä NTPro 5000® Bridge Simulator, which models realistic maritime navigation scenes.
- Three object classes: fixed objects (buoys), targets (moving vessels), and landmasses (coastlines).
- Frames are auto-labelled first, then manually cross-checked for accuracy.
- Frames were sampled at intervals across a ~2-hour recording — not consecutive video frames — to maximize scene diversity.

350 radar frames: 70% train / 20% val / 10% test

7,847 labelled object instances in total

47.6% of instances are landmass — the most common class

Source: Section III-A, Tables 1–2 (Külcü 2026)

A Four-Step Preprocessing Pipeline

1

1. HSV Colour Space

Converts each image from RGB to Hue-Saturation-Value, separating colour from brightness.

2

2. Colour Threshold

Keeps only the narrow yellow hue range that marine radar displays use for strong target echoes.

3

3. Adaptive Binarization

A locally-computed, Gaussian-weighted threshold turns the image into clean, high-contrast black-and-white regions.

4

4. Morphological Closing

An 11×11 kernel joins small gaps and fragments in each echo, without merging unrelated targets together.

Result: cleaner, high-contrast echoes with fewer clutter artefacts — a much easier input for any YOLO model to learn from.

Source: Section III-B, Eq. (1)–(4) (Külcü 2026)

Picking the Right Morphology Kernel Size

Kernel sizes from 1×1 to 13×13 were tested to join broken echo fragments. Too small, and echoes stay fragmented; too large, and unrelated echoes merge together.

< 7×7

Too small

Leaves gaps — echoes stay broken into fragments, causing missed or split detections.

11×11

Just right

Bridges small gaps and connects fragmented land or coastline reflections, without over-merging separate targets.

> 11×11

Too large

Oversmooths detail and risks merging genuinely separate, independent echoes into one.

Source: Section III-B-4, Fig. 2 (Külcü 2026)

Turning Contours into Bounding-Box Labels

For every detected contour, the bounding-box area decides its class — a simple, reproducible way to bootstrap annotations before manual review:

Fixed (buoys)

Area < T_fixed — small, stationary objects

Target (vessels)

T_fixed ≤ Area < T_target — moderately sized, dynamic objects

Landmass (coastline)

Area ≥ T_target — large, stable regions

Important caveat, stated by the author:

Area alone can't capture range or aspect-angle effects, so this is only an initial annotation aid — every automatically generated label is manually cross-validated and corrected afterward.

Source: Section III-C, Eq. (5), Algorithm 2 (Külcü 2026)

Five YOLO Generations, Head to Head

Model	Key Idea
YOLOv5s	Lightweight, anchor-based baseline — solid, but anchors don't suit maritime targets' variable shapes well.
YOLOv8s	Anchor-free detection with decoupled heads — better at small, scattered, low-contrast targets.
YOLOv10s	Efficient backbone (ELAN) for resource-constrained, real-time environments.
YOLOv11s	Builds on v8s with a refined backbone/neck and improved training for better generalization.
YOLOv12s	Attention-centric design (R-ELAN, Area Attention) — the newest, most accuracy-focused variant tested.

Source: Section III-D (Külcü 2026)

Seeing Smaller: The P2 Feature Layer

What's the problem?

- Standard YOLOv12s looks at feature maps at scales P3, P4, and P5 — each one coarser (lower resolution) than the last.
- For tiny objects like buoys, even the finest of these scales can be too coarse to keep enough detail.
- The fix: add a new, higher-resolution P2 layer — a quarter of the input resolution — feeding directly into the detection head.

Detection head now covers 4 scales

P2 / 4 (new — high-res, for small objects)

P3 / 8

P4 / 16

P5 / 32

All the attention-based building blocks of YOLOv12s (R-ELAN, Area Attention) stay in place — P2 is an addition, not a replacement.

Source: Section III-D-1, Eq. (8)–(10), Fig. 4 (Külcü 2026)

Teaching the Model to Care About Small Objects

The idea

- During training, small bounding boxes naturally contribute a weaker signal to the loss — so the model can end up under-learning them.
- The fix: double the loss weight for any ground-truth box smaller than 8×8 pixels — about the median size of a buoy in this dataset.
- A weight of 2.0 was chosen empirically: higher values over-focus on small targets and start to hurt accuracy on normal-sized objects.

$$w_i = 2.0 \ \text{ if box area} < 8\times 8 \text{ px}$$
$$w_i = 1.0 \ \text{ otherwise}$$

What this changes

The weight is applied to both the IoU loss and the Distribution Focal Loss for small boxes — pushing the model to fit them more tightly, without touching how larger vessels or landmasses are trained.

Source: Section III-D-2, Eq. (11)–(12) (Külcü 2026)

Baseline: Five Models on Raw, Unprocessed Data

Overall performance (all 3 classes combined) for each model, trained and tested with no preprocessing at all.

Model	Precision	Recall	mAP50	mAP50-95
YOLOv5s	0.748	0.820	0.810	0.663
YOLOv8s	0.803	0.837	0.840	0.685
YOLOv10s	0.717	0.766	0.788	0.637
YOLOv11s	0.770	0.804	0.809	0.662
YOLOv12s	0.810	0.836	0.850	0.686

YOLOv12s leads overall — but fixed objects (buoys) are the clear weak point for every model: even the best score only 0.321 mAP50-95 on that class alone. Small, low-contrast targets are the main limiting factor of the raw-data baseline.

Source: Table 3, Section IV-C (Külcü 2026)

Preprocessing Lifts Every Model — Especially Buoys

Same 5 models, same dataset — now trained and tested on the preprocessed (clutter-reduced) radar frames.

Model	Precision	Recall	mAP50	mAP50-95
YOLOv5s	0.859	0.890	0.902	0.742
YOLOv8s	0.887	0.891	0.910	0.762
YOLOv10s	0.815	0.859	0.885	0.742
YOLOv11s	0.839	0.878	0.884	0.739
YOLOv12s	0.886	0.904	0.917	0.748

The fixed-object (buoy) class sees the biggest jump: for YOLOv12s, mAP50 rises from 0.576 to 0.772, and mAP50-95 rises from 0.321 to 0.503 — simply from cleaning up the input image, with no architecture change at all.

Source: Table 4, Section IV-D (Külcü 2026)

Adding P2 + Area-Aware Loss: The Full Recipe

YOLOv8s and YOLOv12s, tested on preprocessed data, with the P2 layer alone, and then with P2 + the area-aware loss together.

Configuration	Precision	Recall	mAP50	mAP50-95
YOLOv8s + P2	0.898	0.942	0.941	0.789
YOLOv12s + P2	0.901	0.933	0.946	0.807
YOLOv8s + P2 + Loss	0.895	0.967	0.957	0.813
YOLOv12s + P2 + Loss	0.889	0.964	0.958	0.812

Both models converge to nearly identical, excellent results: around 0.958 mAP50 and 0.812 mAP50-95. The area-aware loss mainly boosts recall — it makes sure weak, buoy-like echoes are no longer ignored during training.

Source: Table 5, Section IV-E (Külcü 2026)

4.F–G · RESULTS

Is All This Accuracy Worth the Extra Computation?

~36 ms	**~30 ms**	**+2–8 ms**
avg. inference time for baseline models on raw images	avg. inference time on preprocessed images (smaller, cleaner input)	extra latency for P2-enhanced models — the accuracy/speed trade-off

Lightweight backbones as an alternative

- GhostNet, MobileNetV3, EfficientNet-B0, and ShuffleNetV2 were also tested on the preprocessed data as lighter alternatives to full YOLO models.
- ShuffleNetV2 is the fastest and lightest (1.8 GFLOPs), at a small accuracy cost.
- GhostNet gives the best balance among the lightweight options — competitive mAP with far fewer parameters than a full YOLO model.

Source: Tables 6–9, Sections IV-F–G (Külcü 2026)

The Key Test: Does This Work on Real Radar Data?

All previous results used simulated radar. Here the exact same models are tested — without retraining — on real, ship-captured radar from the DAAN dataset (Baltic Sea):

Configuration	TP	FP	FN	mAP50	mAP50-95
Raw (baseline)	3301	41	1339	0.698	0.675
Preprocessed	4296	230	344	0.915	0.724
Preproc. + P2	4306	315	334	0.917	0.741
Preproc. + P2 + Loss	**4329**	**315**	**311**	**0.922**	**0.745**

Missed detections (FN) drop from 1,339 to 311 — a 77% reduction. The gains found on simulated data transfer to real radar imagery: mAP50 climbs from 0.698 to 0.922, confirming this isn't just an artefact of the simulator.

Source: Table 10, Section IV-H (Külcü 2026)

CONCLUSION

Preprocessing and Architecture Are Complementary

- Domain-specific preprocessing alone closes most of the gap on small, low-contrast buoys — the hardest class in the raw-data baseline.
- The P2 layer and area-aware loss add a further, targeted boost to small-object sensitivity, without hurting performance on vessels or landmasses.
- Crucially, these gains hold up on real DAAN radar data, not just the simulator — evidence of genuine cross-domain generalization.
- Future work: collect larger real-world radar datasets from multiple platforms, and explore domain-adaptation methods to generalize across different radar systems.

REFERENCE

Source

Külcü, S. (2026).

Enhancing Radar PPI-Based Maritime Object Detection Using Hybrid Preprocessing and Deep Learning Approach.

IEEE Access, 14, 111715–111730. doi:10.1109/ACCESS.2026.3715472

External validation dataset: DAAN real-world marine radar repository, German Aerospace Center (DLR)

This presentation is an academic summary of the research paper above. For the full reference list (46 citations), please see the original article.

Thank You