Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.6, December 2025, pp. 3648~3656
e-ISSN: 2548-6861

Improving Efficient Ship Detection Performance Using Contextual Transformers for Maritime Surveillance

Marsel Marhaen Wungow[1], Dayen Manoppo[2], Ni Made S. Mustikayani[3], Muhamad Dwisnanto Putro[4*]
[1,2,3]Jurusan Teknik Elektro, Program Studi Teknik Informatika, Universitas Sam Ratulangi, Kota Manado
[4]Program Master Informatika, Program Pascasarjana, Universitas Sam Ratulangi, Kota Manado
marselwungow026@student.unsrat.ac.id[1], dayenmanoppo026@student.unsrat.ac.id[2], nimustikayani026@student.unsrat.ac.id[3], dwisnantoputro@unsrat.ac.id[4*]

Article Info

Article history:

Received 2025-09-12
Revised 2025-11-20
Accepted 2025-11-26

Keyword:

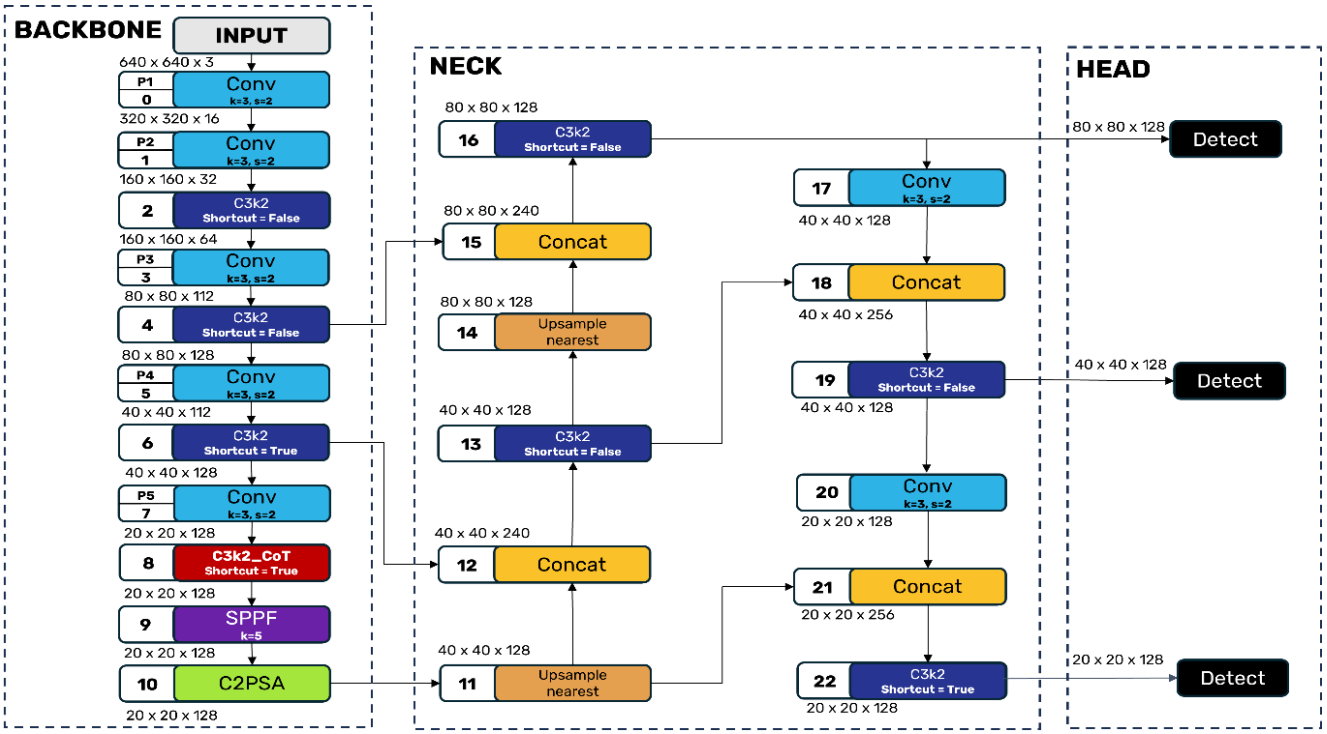
Ship detection,
Deep Learning,
Modified YOLO11,
Contextual Transformer,
Efficient Model.10

ABSTRACT

Ship surveillance plays a crucial role in enhancing defense systems in coastal areas. An automatic vessel detection system is necessary to accurately identify vessels and their categories, typically utilizing a reliable computer vision system. The nano version of YOLO11 has emerged as one of the object detection methods that officially provides lightweight computing, but still has limitations in extracting complex features. Contextual Transformer (CoT) efficiently utilizes long-range relationships, thereby enhancing feature discrimination performance. This study proposes a vessel detection system by modifying the YOLO11 architecture using the Contextual Transformer block. This work introduces YOLO11-Pico, a lighter version of nano, with channel size adjustments at certain stages for further efficiency. The proposed CoT block applies fewer multiplication mapping operations, which are capable of representing global features to obtain richer contextual information. The SeaShips dataset is used as the source of data for model training and evaluation. Experimental results demonstrate that the proposed model YOLO11-pico-CoT achieves superior performance compared to prominent lightweight YOLO architectures, including the YOLO11n baseline, YOLOv5n, YOLOv10n, and the latest YOLOv12n. The integration of CoT contributes positively to improving the accuracy of ship category and location predictions, achieving 0.964 mAP50 and 0.714 mAP50:95. Additionally, efficiency evaluations show that the proposed module is computationally lighter and has fewer parameters, specifically 1,711,250 parameters while operating at 3.97 FPS, giving it an advantage in terms of capabilities over the comparison methods.

I. PENDAHULUAN

Penerapan sistem deteksi berbasis teknologi telah memberikan dampak signifikan di berbagai sektor, termasuk pendidikan, kesehatan, transportasi, dan kemaritiman. Sistem deteksi ini berperan penting dalam meningkatkan efisiensi kerja manusia, khususnya dalam aktivitas pengawasan dan pemantauan, sehingga mendukung operasional yang lebih akurat dan responsif. Sistem konvensional seperti *Automatic Identification System* (AIS) dan radar masih memiliki keterbatasan dalam mendeteksi kapal tidak dilengkapi *transponder* aktif [1]. Hal ini mendorong kebutuhan mendesak akan sistem deteksi visual yang lebih adaptif, presisi, dan dapat diimplementasikan pada perangkat dengan sumber daya terbatas. Implementasi deteksi visual di domain maritim menghadapi tantangan unik yang secara signifikan menurunkan akurasi. Tantangan-tantangan ini meliputi variasi skala kapal yang ekstrem, mulai dari kapal kecil di kejauhan hingga kapal besar yang mendominasi citra. Selain itu, masalah oklusi antar kapal di pelabuhan yang ramai, serta latar belakang laut yang cenderung homogen atau monoton, menyulitkan model untuk membedakan fitur objek secara efektif [2], [3], [4]. YOLO (*You Only Look Once*) [26]



Gambar 1. Arsitektur YOLO11. Algoritma ini mempekerjakan Backbone, Neck, dan Head yang saling bekerja sama untuk mengekstraksi informasi dan melokalisasi objek

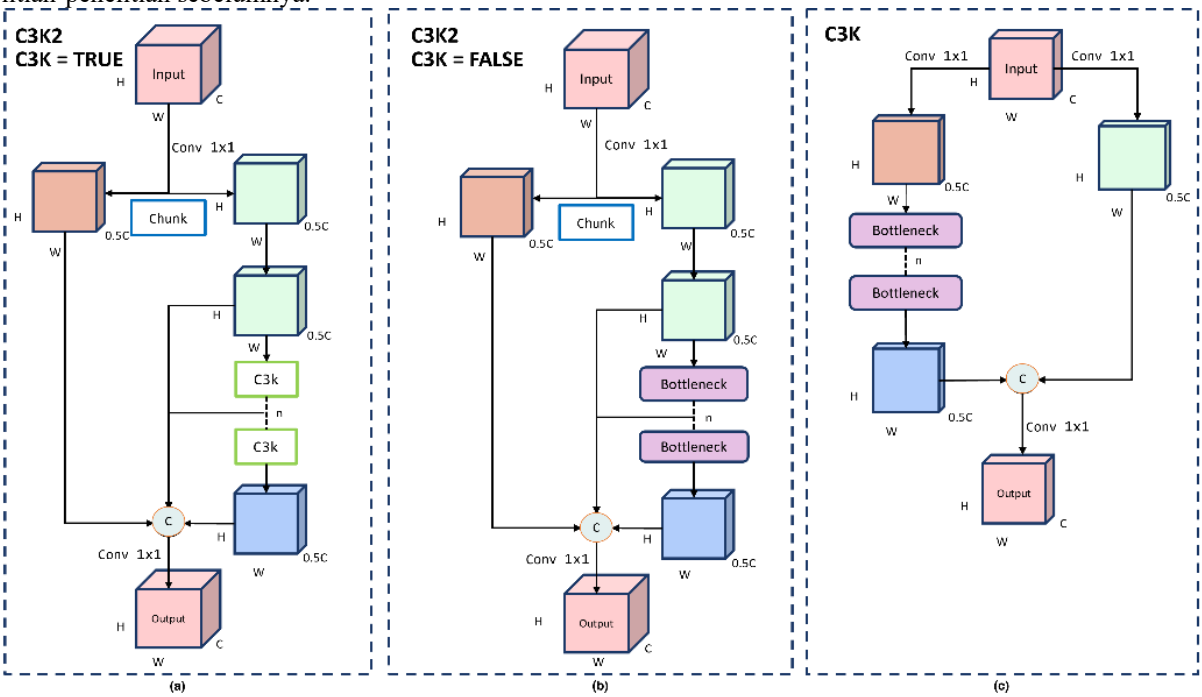
merupakan arsitektur deteksi objek secara visual yang melibatkan proses satu tahap secara cepat dan akurat. YOLO11 menawarkan model dengan efisiensi tinggi, termasuk versi YOLO11-nano yang dirancang khusus untuk kebutuhan edge computing [2]. Versi ringan ini memiliki parameter kecil untuk implementasi di perangkat berbiaya rendah, sehingga unggul dalam hal efisiensi komputasi dan kecepatan inferensi. Namun, arsitektur yang ringan ini memiliki keterbatasan utama yang menjadi fokus penelitian ini. Ukuran model yang kecil secara langsung berdampak pada penurunan kapasitas ekstraksi fitur, yang mengakibatkan penurunan akurasi deteksi (performa), terutama dalam menangani fitur-fitur kompleks di lingkungan maritim. Oleh karena itu, tantangan utamanya adalah meningkatkan akurasi YOLO11-nano tanpa mengorbankan efisiensi komputasinya. Untuk mengatasi hal tersebut, metode *Transformer* menjadi solusi dalam memperkuat representasi global melalui mekanisme *self-attention* [5], [6], [7], [8].

Meskipun akurat, mekanisme ini memiliki tantangan dalam beban komputasi yang tinggi. *Contextual Transformer* (CoT) hadir sebagai alternatif efisien, yang dapat mempercepat proses inferensi, tanpa mengorbankan performa diskriminasi fitur [7], [9], [16]. CoT juga terbukti mengatasi tantangan multiskala dan oklusi di lingkungan maritim [6], [10].

Berbagai model telah dikembangkan untuk meningkatkan performa di sektor kemaritiman. YOLO-HPSD [2] mengusulkan piramida fitur adaptif untuk mendeteksi kapal pada berbagai skala, sedangkan YOLO-LPSS [3] memperbaiki sensitivitas terhadap kapal kecil dengan penyesuaian strategi anchor. EGM-YOLOv8 [4] menggabungkan fitur lokal dan global secara efisien, dan CM-YOLO [5] mengadopsi modul konteks yang ringan untuk menghadapi latar belakang yang kompleks. Sementara itu, DSONet [6] dengan *dual-scale occlusion-aware* untuk mengatasi oklusi antarkapal. Mayoritas berfokus pada peningkatan performa, dan komputasi tinggi seperti EGM-YOLOv8 dan CM-YOLO [4], [5]. Model ringan seperti YOLO-HPSD [2], YOLO-LPSS [3], dan DSONet [6] menekankan efisiensi dan kecepatan untuk mendukung penerapan pada perangkat berbiaya rendah yang berjalan secara *real-time*.

Tantangan tambahan seperti visibilitas rendah dan ukuran objek kecil turut mendorong inovasi dalam arsitektur. Pekerjaan *Spotlighting* [8] menerapkan *image dehazing* untuk memperbaiki kejernihan citra, sedangkan *Adaptive Head Pruning* [9] membantu memangkas latensi dan ukuran model tanpa menurunkan akurasi. Evaluasi dilakukan pada *dataset* seperti DAShip [10] dan SeaShips [11], [12]. Model-model terbaru seperti AFF-LightNet [11], YOLO-StarLS [12], Improved YOLOv8n [13], YOLOv7-Ship [21], dan LH-YOLO [24] fokus pada efisiensi tanpa mengorbankan akurasi [17], [25]. Pendekatan CNN, seperti ResNet dengan CBAM [19] menunjukkan akurasi yang meningkat secara signifikan.

Selain itu, YOLOv7oSAR [18] dirancang untuk deteksi kapal dalam citra radar SAR dengan akurasi tinggi, yang memperkuat pendekatan CNN dalam domain non-optik [22]. Dengan mempertimbangkan keterbatasan perangkat keras serta tuntutan aplikasi waktu nyata pada perangkat berbiaya rendah, deteksi kapal yang ringan, efisien, dan presisi menjadi sangat penting dalam mendukung pengawasan maritim yang adaptif dan berkelanjutan [20], [23]. Oleh karena itu, kondisi ini membentuk celah penelitian (*research gap*) dari studi yang diusulkan dibandingkan dengan penelitian-penelitian sebelumnya.



Gambar 2. Blok C3K2 dengan C3K (a), Blok C3K2 dengan Botleneck (b), Blok C3K (c)

Penelitian ini memiliki dua tujuan spesifik yang saling terkait. Pertama, untuk menganalisis trade-off antara akurasi deteksi dan efisiensi komputasi saat mengintegrasikan modul transformer ke dalam arsitektur lightweight YOLO11-nano. Kedua, untuk menyelidiki secara spesifik bagaimana peningkatan representasi fitur global melalui Contextual Transformer dapat meningkatkan presisi deteksi tanpa memberikan beban komputasi yang masif, yang menjadi tantangan utama pada model lightweight.Adapun kontribusi utama dari penelitian ini, dapat dirangkum sebagai berikut:

- Merancang arsitektur hibrid baru, YOLO11-pico-CoT, yang mengintegrasikan blok adaptasi CoT, Modifikasi Bottleneck, dan CRB ke dalam baseline YOLO11-pico yang ringan.
- Analisis mendalam mengenai efek Contextual Transformer terhadap representasi spasial. Analisis ini secara kualitatif menunjukkan bagaimana CoT meningkatkan kemampuan model dalam menangkap informasi kontekstual objek kapal.
- Analisis komprehensif mengenai efisiensi memori dan komputasi untuk mengukur trade-off dari penambahan blok CoT, yang membuktikan kelayakan model pada perangkat berbiaya rendah.

II. METODE

Arsitektur YOLO11 tersusun atas tiga komponen fundamental yang bekerja secara sinergis untuk meningkatkan kinerja deteksi objek, yaitu Backbone, Neck, dan Head. Gambar 1 menunjukkan arsitektur yang diusulkan, yang mengintegrasikan inovasi terkini untuk memastikan efisiensi dan akurasi yang optimal. *Backbone* berfungsi sebagai ekstraktor fitur utama dari citra *input*. Pada YOLO11, digunakan serangkaian lapisan konvolusi yang diterapkan dalam blok C3k2 [29], sebuah pengembangan dari blok C3 tradisional yang mengurangi beban komputasi

TABEL I
PERBEDAAN ANTARA YOLO11-NANO DAN YOLO11-PICO

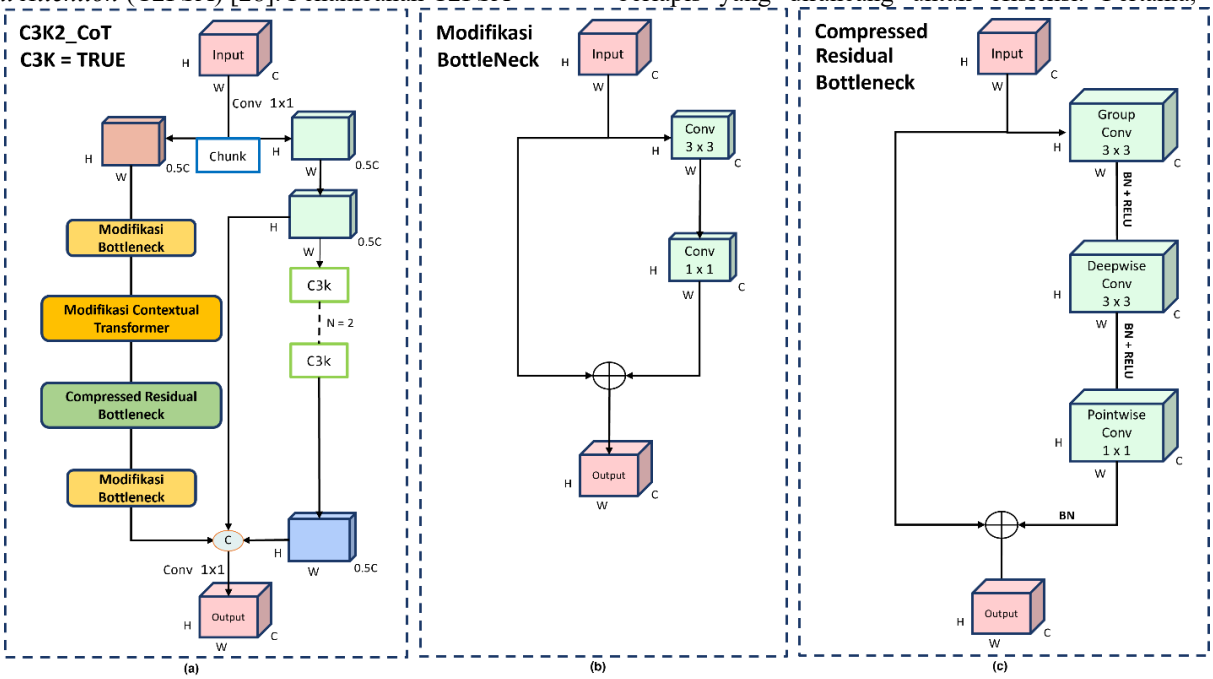
Komponen	YOLO11-nano	YOLO11-pico
Jumlah banyaknya saluran	4403	3507

dengan meminimalkan operasi perkalian tanpa mengorbankan kapasitas representasi [30].

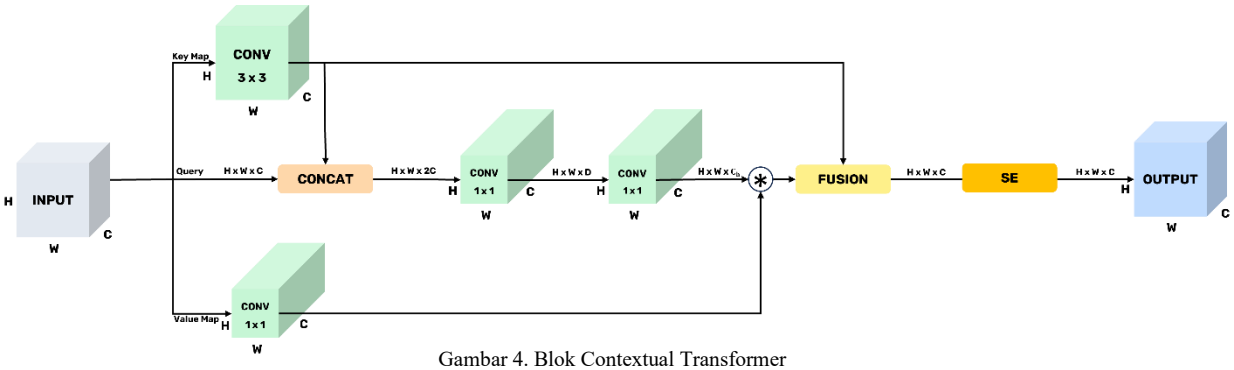
Pada C3k2 memiliki dua kondisi yaitu *True* dan *False*. *True* memakai C3k yang ditunjukkan pada Gambar 2(c).

Mekanisme *BottleNeck* menghemat sumber daya komputasi, seperti yang ditunjukkan pada Gambar 2(a). Sedangkan kondisi *False* memakai *BottleNeck* yang ditunjukkan pada Gambar 2(b). Setelah itu, modul SPPF (*Spatial Pyramid Pooling–Faste*r) [27], menangkap informasi pada berbagai skala yang kaya akan informasi spasial dan tekstural. Modul ini memungkinkan model untuk mendeteksi objek pada skala besar maupun kecil. Pada *backbone* ini, YOLO11 juga menambahkan blok baru, yaitu *Cross Stage Partial with Spatial Attention* (C2PSA) [28]. Penambahan C2PSA berhasil mengurangi jumlah saluran sebanyak 896, yang secara langsung berdampak pada penurunan signifikan jumlah parameter dan beban komputasi. Konsekuensinya, arsitektur YOLO11-pico dirancang untuk memberikan kinerja deteksi yang optimal pada perangkat berbiaya rendah atau perangkat dengan sumber daya komputasi terbatas. Perbandingan rinci mengenai modifikasi arsitektur ini disajikan pada Tabel 1.

Penelitian ini mengusulkan kontribusi arsitektural berlapis yang dirancang untuk efisiensi. Pertama, kami



Gambar 3. Blok C3K2 dengan Blok Modifikasi Bottleneck, Blok CoT, Blok CRB (a), Blok Modifikasi Botleneck (b), Blok CRB (c)



Gambar 4. Blok Contextual Transformer

memberikan mekanisme perhatian spasial yang memungkinkan model untuk lebih efektif memusatkan fokus pada wilayah penting dalam citra. Studi ini memperkenalkan YOLO11-pico, sebuah arsitektur model yang dioptimalkan untuk efisiensi komputasi sebagai turunan dari model YOLO11-nano. Peningkatan efisiensi ini dicapai melalui perampingan arsitektur, secara spesifik dengan membatasi jumlah maksimum saluran pada peta fitur. Modifikasi ini memperkenalkan dua blok komputasi ringan baru, yaitu Modifikasi Bottleneck dan Compressed Residual Bottleneck (CRB). Kedua, kami mengadaptasi blok Contextual Transformer (CoT) [14]. Adaptasi ini mencakup penyesuaian hiperparameter internal, seperti ukuran kernel dan reduction factor, untuk mengoptimalkan keseimbangan antara representasi fitur kontekstual dan efisiensi komputasi.

Inovasi utama kami terletak pada perancangan blok hibrid baru yang mengintegrasikan ketiga komponen tersebut ke dalam jalur bypass C3k2, seperti diilustrasikan pada Gambar 3(a). Blok hibrid ini memiliki tumpukan sekuensial yang dirancang khusus, yang terdiri dari urutan: Modifikasi Bottleneck, diikuti oleh CoT, kemudian CRB, dan diakhiri dengan Modifikasi Bottleneck. Perancangan ini secara spesifik bertujuan untuk menyeimbangkan ekstraksi fitur kontekstual dari CoT dengan efisiensi komputasi yang tinggi dari blok CRB dan Modifikasi Bottleneck yang kami usulkan.

A. *Contextual Transformer*

Contextual Transformer (CoT) memperkenalkan pendekatan baru dengan memanfaatkan konvolusi lokal untuk mengekstraksi informasi kontekstual [14], seperti yang diilustrasikan pada Gambar 4. Studi yang diusulkan menempatkan modul CoT pada blok C3k2, yang diintegrasikan di Backbone untuk memperkuat representasi informasi kontekstual. Penempatan ini tepatnya pada layer 8, seperti yang diilustrasikan pada Gambar 1. Di dalam blok C3k2 itu sendiri, modul CoT ditempatkan pada bagian fitur bypass, seperti yang ditunjukkan pada Gambar 3a. Proses dimulai dengan mengekstraksi *Key*, *Query*, dan *Value* dari masukkan peta fitur. Pendekatan ini dapat dijelaskan melalui ilustrasi berikut:

TABEL II
EVALUASI PERFORMA MODEL DAN KOMPARASINYA DENGAN METODE PEKERJAAN TERKAIT

Model	Parameter	mAP50	mAP50:95	Precision	Recall
YOLOv5n	2,509,634	0,958	0,700	-	-
YOLOv6n	4,238,738	0,959	0,718	-	-
YOLOv8n	3,012,018	0,963	0,725	0,950	0,922
YOLOv10n	2,709,380	0,938	0,699	0,916	0,880
YOLO11n	2,591,010	0,960	0,705	0,955	0,911
YOLOv12n	2,539,266	0,959	0,699	0,942	0,913
YOLOv10-HPSD [2]	2,800,000	0,988	-	0,981	0,976
DSONet [6]	1,800,000	0,949	0,689	0,933	0,889
YOLOv7-Ship [21]	6,100,000	0,805	0,554	0,814	0,758
YOLO11-pico	1,522,904	0,959	0,703	0,948	0,905
YOLO11-pico-CoT (diusulkan)	**1,711,250**	**0,964**	**0,714**	0,943	0,919

$$Q, K = X, \tag{1}$$

$$V = W_v(X). \tag{2}$$

Query dan *Key* didefinisikan sebagai masukkan fitur X. Sementara itu, *Value* merupakan representasi dari masukkan peta fitur yang telah dilakukan konvolusi (W) dengan filter berukuran 3×3. Konvolusi grup digunakan untuk memodelkan area lokal pada peta fitur *Key* secara efisien dengan menggunakan filter yang dapat disesuaikan. Fitur *Query* tetap diteruskan untuk memperkaya representasi dengan meningkatkan variasi fitur. Proses ini dapat dijelaskan melalui ilustrasi berikut:

$$B_1 = W_\delta(W_\theta([K_G, Q])), \tag{3}$$

dimana,

$$K_G = GConv(K). \tag{4}$$

K_G menjadi representasi yang telah di kontekstualisasi menggunakan operasi grup konvolusi (GConv) dari masukkan X. Setelah itu, dengan menggabungkan K_G dan *Query*, matriks perhatian dibentuk melalui dua tahapan konvolusi 1×1 secara berurutan, di mana konvolusi pertama W_θ menggunakan aktivasi GELU, sedangkan konvolusi kedua W_δ tidak menggunakan fungsi aktivasi. Peta fitur perhatian B_1 yang telah terbentuk kemudian digunakan untuk menggabungkan informasi dari *Value*, yang sebelumnya diperoleh melalui konvolusi 1×1 dari masukan. Proses ini dapat diilustrasikan sebagai berikut:

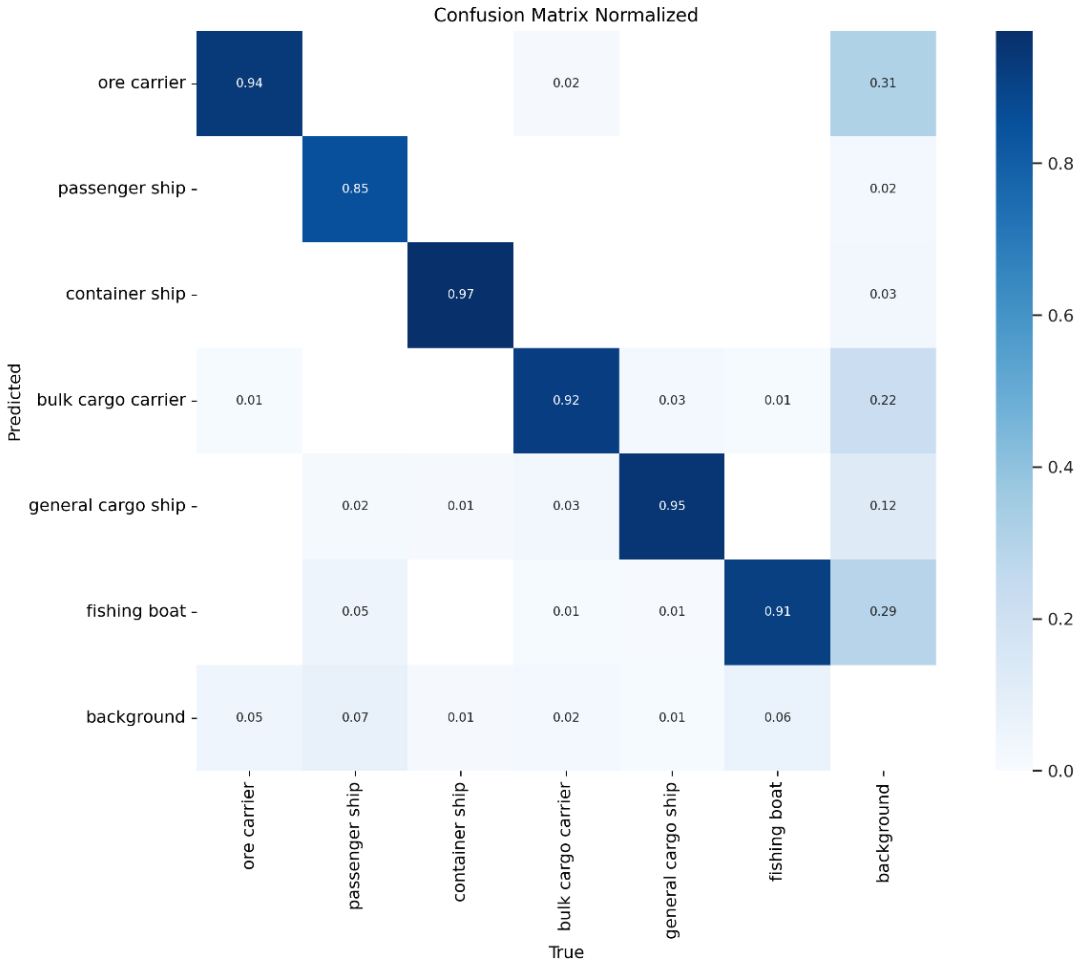
$$B_2 = V * B_1. \tag{5}$$

TABEL III
ANALISA EFISIENSI MODEL

Model	GFLOPs	Parameter	FPS
YOLOv5n	7,2	2,509,634	4,18
YOLOv6n	11,9	4,238,738	3,37
YOLOv8n	8,2	3,012,018	3,97
YOLOv10n	8,4	2,709,380	3,39
YOLO11n	6,4	2,591,010	4,04
YOLOv12n	6,0	2,539,266	3,50
YOLO11-pico	5,8	1,522,904	4,41
YOLO11-pico-CoT	**6,3**	**1,711,250**	**3,97**

Peta fitur B_2 digabungkan dengan informasi dari peta fitur K_G, menggunakan Squeeze and Excitation blok [15]. Menggunakan nilai reduksi setengah dari peta fitur masukkan, dan memakai aktivasi GELU. Proses ini menghasilkan representasi akhir yang lebih informative, seperti yang dapat diekspresikan sebagai berikut:

$$CoT = B_2 + K_G. \tag{6}$$

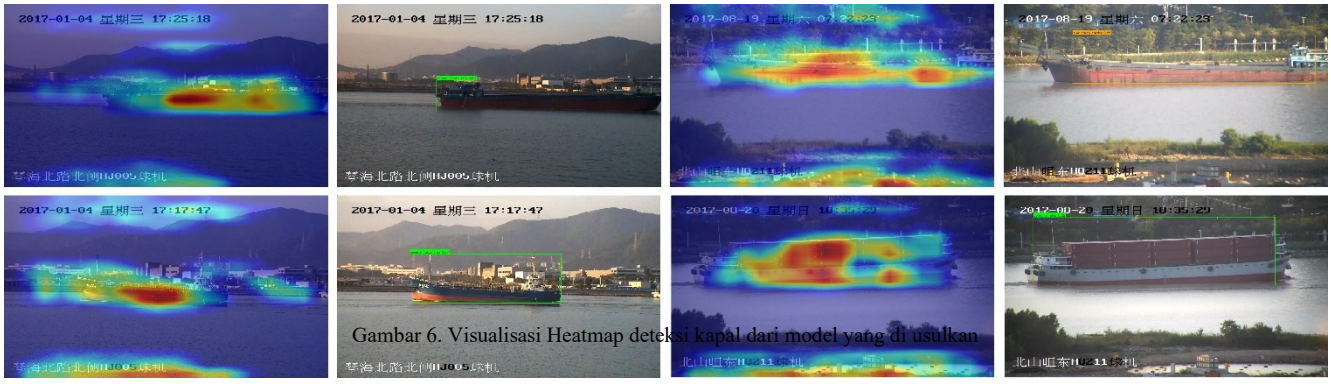


B. Modifikasi Bottleneck

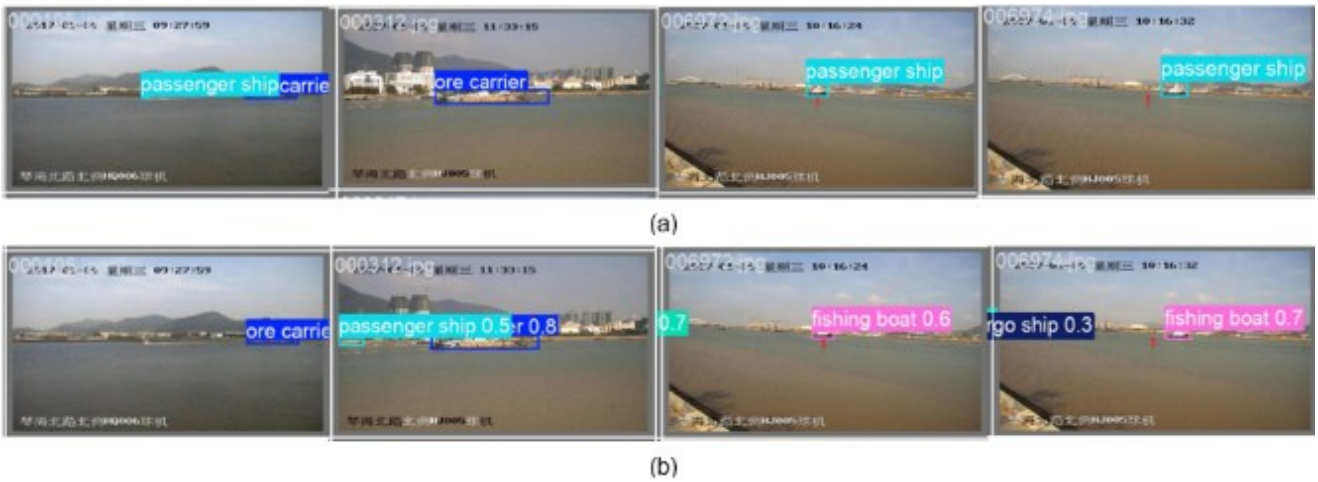
Bottleneck merupakan blok yang terdiri atas dua proses sekuensial convolution yang masing-masing dilengkapi dengan operasi konvolusi, batch normalization, dan aktivasi SiLU. Blok ini mengekstraksi fitur masukan secara efisien dengan memanfaatkan mekanisme residual. Conv1 berfungsi untuk mereduksi jumlah peta fitur, kemudian jumlah tersebut dikembalikan lagi melalui operasi Conv2. Modifikasi dilakukan pada ukuran kernel Conv2, yaitu dari 3×3 menjadi 1×1 seperti yang ditunjukkan pada gambar 3(b). Tujuan penggantian kernel size tersebut adalah untuk menyederhanakan kompleksitas komputasi sekaligus meningkatkan akurasi. Pada pengembangan yang dilakukan, penggunaan kernel 1×1 terbukti mampu menghasilkan peningkatan akurasi dibandingkan kernel 3×3.

C. Compressed Residual Bottleneck

Compressed Residual Bottleneck (CRB) merupakan modifikasi ringan dari bottleneck ResNet yang menggantikan konvolusi 1×1 dengan grouped convolution untuk



Gambar 6. Visualisasi Heatmap deteksi kapal dari model yang di usulkan



Gambar 7. Visualisasi kasus kesalahan deteksi model yang diusulkan. (a) Ground truth objek. (b) Prediksi model dengan label dan bounding box yang keliru.

meningkatkan efisiensi tanpa mengorbankan kualitas representasi. seperti yang diilustrasikan pada Gambar 3(c) Studi yang diusulkan menempatkan modul CRB pada blok C3k2 yang ditunjukkan pada Gambar 3(a). Blok ini terlebih dahulu mengompresi jumlah kanal dengan grouped convolution untuk menekan kompleksitas, kemudian menggunakan depthwise convolution dalam ekstraksi fitur spasial dan pointwise convolution untuk mengembalikan kanal ke ukuran semula. Setiap tahap diperkuat dengan batch normalization dan aktivasi SiLU guna menjaga stabilitas pelatihan, sementara residual connection memastikan informasi fitur asli tetap terjaga. Oleh karena itu, CRB mampu memperkaya representasi fitur secara efisien, sehingga efektif diterapkan pada skenario lightweight dan real-time.

III. TRAINING KONFIGURASI

A. Dataset

Dataset pada penelitian ini menggunakan SeaShips [11]. Dataset maritim ini menyediakan 7,000 gambar dengan resolusi 1920×1080 piksel, mencakup enam kategori kapal yaitu, ore carrier, bulk cargo carrier, general cargo ship, container ship, fishing boat, dan passenger ship dengan anotasi kotak pembatas yang presisi. Dataset ini secara inheren mencakup berbagai tantangan domain maritim yang disoroti dalam penelitian, seperti variasi ukuran kapal yang signifikan, oklusi parsial, dan latar belakang yang kompleks.

Untuk pelatihan dan evaluasi, *dataset* ini dibagi menjadi 50% untuk pelatihan (1,750 citra), 25% untuk validasi (1,750 citra), dan 25% untuk pengujian (3,500 citra).

B. Konfigurasi Pelatihan dan Pengujian

Penelitian dilakukan menggunakan *platform* komputasi awan Kaggle Notebook dalam sebuah lingkungan virtual yang dilengkapi dengan prosesor Intel(R) Xeon(R) CPU @ 2,00GHz, akselerator GPU NVIDIA Tesla P100-PCIE-16GB dengan 16 GB VRAM, dan memori sistem (RAM) sebesar 32 GB. Modul yang diusulkan dalam penelitian ini menggunakan *framework* PyTorch versi 2,6,0 dan memanfaatkan CUDA versi 12,5 untuk efisiensi akselerator GPU. Proses pada *training* dilakukan sebanyak 200 *epoch*, dengan *batch size* sebesar 16. Percobaan ini juga melibatkan image size berukuran 640×640, learning rate sebesar 0,01 dan menggunakan optimizer SGD.

Untuk meningkatkan robustitas model terhadap variasi iluminasi dan artefak citra, kami menerapkan augmentasi berbasis Albumentations yang terintegrasi dalam konfigurasi pelatihan YOLO termasuk Blur, MedianBlur, transformasi grayscale, CLAHE, dan augmentasi geometris seperti mosaic dengan probabilitas aplikasi yang rendah (p = 0,01) untuk menambah keragaman kondisi pencahayaan tanpa memperkenalkan noise berlebih; augmentasi ini diaktifkan selama sebagian besar proses pelatihan dan dinonaktifkan pada 10 epoch terakhir agar model dapat melakukan penyetelan akhir pada data asli, sehingga membantu stabilitas konvergensi dan meningkatkan generalisasi di lingkungan maritim yang beragam.

IV. HASIL DAN PEMBAHASAN

Evaluasi kuantitatif dalam penelitian ini berfokus pada dua kategori utama: kinerja akurasi dan efisiensi model. Kinerja akurasi, yang mengukur keandalan deteksi, dievaluasi menggunakan mean Average Precision (mAP) sebagai metrik utama yang dihitung dari kurva Precision-Recall. Metrik ini didukung oleh Precision, yang mengukur proporsi deteksi benar dari total prediksi yang dihasilkan, dan Recall, yang mengukur proporsi deteksi benar dari total objek aktual yang ada (ground truth). Sementara itu, kinerja efisiensi mengukur kelayakan implementasi model pada perangkat berdaya rendah, yang dievaluasi melalui tiga metrik: Jumlah Parameter yang merepresentasikan ukuran model, GFLOPs (Giga Floating Point Operations per Second) yang mengukur kompleksitas komputasi, dan FPS (Frames

Per Second) yang mengukur kecepatan inferensi untuk kelayakan real-time.

A. Evaluasi Model Pada Dataset

Evaluasi kinerja komprehensif dilakukan melalui analisis komparatif terhadap arsitektur YOLO seri nano terkini. Rangkaian model pembanding ini meliputi baseline YOLOIIn, beserta YOLOv5n, YOLOv6n, YOLOv8n, YOLOv10n [31], dan YOLOv12n [32]. Evaluasi ini berfokus pada kelayakan implementasi untuk perangkat berbiaya rendah. Kinerja model diukur menggunakan dua kriteria utama. Kriteria pertama adalah akurasi deteksi, yang dikuantifikasi melalui mean Average Precision atau mAP. Kriteria kedua adalah efisiensi model, yang ditentukan oleh jumlah parameter. Perbandingan kuantitatif dari kedua metrik tersebut disajikan secara rinci pada Tabel II.

YOLOv12n meraih akurasi mAP50:95 sebesar 69,9% yang dimana berada dibawah YOLO11n. Hasil kuantitatif juga menunjukkan bahwa model YOLO11n mengungguli YOLOv5n. Dibandingkan model seperti YOLOv8n, YOLOv6n dan YOLOv12n model YOLO11n memiliki performa dan parameter yang paling optimal yaitu dengan yang lebih sedikit dengan performa yang dapat bersaing. Ini juga menunjukkan bahwa YOLO11n memiliki performa yang lebih baik dari versi terbaru YOLOv12n. Pengembangan model yang dilakukan menghasilkan model yang efisien (YOLO11-pico), yang menghasilkan presisi yang lebih rendah dari YOLOv8n. Model efisien yang diusulkan meraih mAP50:95 sebesar 70,3% . Konfigurasi ini memproduksi jumlah parameter yang tidak masif. Model YOLO11-pico-CoT yang diusulkan mencapai 0,714 untuk mAP50:95, precision 0,943, dan recall 0,919, menunjukkan peningkatan performa sebesar 0,009 dibandingkan baseline YOLO11n yang mencatat 0,705 mAP50:95, precision 0,955, dan recall 0,911. Peningkatan ini menjadi sangat signifikan jika ditinjau bersamaan dengan pengurangan parameter model sebesar 33,95%, yaitu dari 2,59 Juta menjadi 1,71 Juta.

Berdasarkan confusion matrix pada Gambar 5, model deteksi yang diusulkan mendapatkan hasil true positive untuk prediksi jenis kapal ore carrier sebesar 0.95, passenger ship sebesar 0,86, container ship sebesar 0,97, dan bulk cargo carrier sebesar 0,94, general cargo ship sebesar 0,95, dan fishing boat sebesar 0,91 pada dataset test. Model deteksi yang diusulkan bisa membedakan ciri-ciri kapal dengan tepat, sehingga dapat membedakan baik kapal berukuran besar dan kecil. Meskipun demikian, analisis kesalahan, baik secara kuantitatif maupun kualitatif, menunjukkan keterbatasan model yang diusulkan. Secara kuantitatif, confusion matrix pada Gambar 5 menunjukkan performa terendah terjadi pada kategori passenger ship dengan akurasi 0.85, di mana model terkadang keliru mengklasifikasikannya sebagai fishing boat atau background. Keterbatasan ini juga ditunjukkan secara visual pada Gambar 7. Terlihat jelas pada Gambar 7(b), terjadi beberapa jenis kesalahan seperti false negative, di mana ground truth passenger ship pada Gambar 7(a) gagal diprediksi; false positive, di mana model memprediksi passenger ship pada area tidak berlabel; dan kesalahan klasifikasi, di mana label passenger ship keliru diprediksi sebagai fishing boat. Kedua analisis ini mengindikasikan bahwa fitur-fitur pembeda antar kategori kapal, terutama untuk kelas passenger ship, masih memerlukan perbaikan lebih lanjut.

Modul Contextual Transformer yang dimasukkan pada YOLO11-pico, yang dimana memainkan peranan penting dalam menangkap informasi yang menarik. Hal ini dapat divisualisasikan heatmap pada Gambar 6, yang membuktikan bahwa modul yang diusulkan menangkap informasi penting dari objek kapal. Modul perhatian ini merelasikan informasi pasial yang berbeda, meningkatkan variasi cakupan ekstraksi dan menghasilkan bobot terlatih dari setiap informasi. Perbedaan informasi ini menghubungkan fitur objek dengan latar belakang, yang menentukan nilai pentingnya.

B. Analisa Efisiensi Pada Model

Efisiensi model menggambarkan kemampuan pendeteksi untuk diimplementasikan pada aplikasi nyata. Permasalahan ini adanya sistem penglihatan yang dapat beroperasi pada perangkat dengan sumber daya terbatas. Model diuji pada Perangkat Raspberry Pi 5 untuk mengevaluasi kinerjanya. YOLO11-pico menunjukkan efisiensi yang lebih tinggi dibandingkan YOLO11n, YOLOv5n, YOLOv6n, YOLOv8n, YOLOv10n, dan YOLOv12n, sebagaimana ditampilkan pada Tabel 3.

Selanjutnya, YOLO11-pico-CoT mampu mencapai kecepatan inferensi sebesar 3,97 FPS, yang sama dibandingkan YOLOv8n. Selain itu, modul yang diusulkan memiliki jumlah parameter 31,8% lebih rendah dan biaya komputasi 0,1 GFLOPs lebih ringan dibandingkan YOLOv5n. Hasil efisiensi ini menunjukkan keberhasilan dalam pengurangan parameter sebesar 33,95% dibandingkan YOLO11n. Tetapi mengorbankan kecepatan inferensi, yang hasilnya lebih rendah 0,07 FPS dibandingkan YOLO11n. Modul *Contextual Transformer* berkontribusi dalam meningkatkan performa tanpa menambah beban komputasi yang masif. Hasil ini menunjukkan bahwa arsitektur yang diusulkan memiliki efisiensi komputasi dan peningkatan performa yang optimal dibandingkan YOLO11n.

V. KESIMPULAN

Penelitian ini mengusulkan sebuah sistem deteksi kapal, YOLO11-pico-CoT, yang dirancang untuk beroperasi secara real-time dengan mengandalkan arsitektur pembelajaran mendalam yang dirancang secara efisien. Kontribusi ilmiah utama dari penelitian ini berfokus pada analisis mendalam mengenai efisiensi memori/komputasi dan efek representasi spasial dari integrasi Contextual Transformer.

Dari sisi efisiensi memori dan komputasi, model yang diusulkan terbukti lebih ringan. Model hanya membutuhkan jumlah parameter 33,95% lebih sedikit dibandingkan model YOLO11n original. Selain itu, jumlah komputasi yang

dibutuhkan hanya 6,3 GFLOPs, yang 1,56% lebih ringan dari baseline. Dari sisi representasi spasial, penerapan blok Contextual Transformer terbukti mampu memperkaya fitur dan meningkatkan kemampuan model dalam menangkap informasi kontekstual objek kapal.

Keseimbangan yang sukses antara efisiensi tinggi dan representasi fitur yang kaya inilah yang menghasilkan peningkatan performa deteksi. Sistem yang dikembangkan memberikan hasil mAP50:95 sebesar 0,714, sebuah peningkatan 0,009 dari model YOLO11n original. Hasil ini dapat dicapai dengan pengurangan jumlah kanal pada beberapa lapisan konvolusi sehingga ini dapat mengurangi jumlah parameter dan komputasi sekaligus mereduksi redundansi peta fitur yang tidak penting.

DAFTAR PUSTAKA

[1] L. Qian et al., “A new method of inland water ship trajectory prediction based on long short-term memory network optimized by genetic algorithm,” Appl. Sci., vol. 12, no. 8, art. 4073, 2022.

[2] M. Zhu et al., “YOLO-HPSD: A high-precision ship target detection model based on YOLOv10,” PLOS ONE, vol. 20, no. 1, e0321863, Jan. 2025.

[3] L. Shen et al., “YOLO-LPSS: A Lightweight and Precise Detection Model for Small Sea Ships,” J. Mar. Sci. Eng., vol. 13, no. 2, pp. 115–127, Feb. 2025.

[4] Y. Li and S. Wang, “EGM-YOLOv8: A Lightweight Ship Detection Model with Efficient Global–Local Feature Fusion and Attention,” Proc. Int. Conf. Image Process., Oct. 2025, pp. 543–550.

[5] L. Min, F. Dou, Y. Zhang, D. Shao, L. Li and B. Wang, "CM-YOLO: Context Modulated Representation Learning for Ship Detection," in IEEE Transactions on Geoscience and Remote Sensing, vol. 63, pp. 1-14, 2025, Art no. 4202414, doi: 10.1109/TGRS.2025.3538848.

[6] L. Shen et al., “DSONet: A Dual-Scale Occlusion-Aware Framework for Multiscale Ship Detection,” IEEE Trans. Multimedia, vol. 27, no. 3, pp. 1023–1033, Mar. 2025.

[7] Y. Kumar and P. Lee, “Adaptive Head Pruning for Multi-Head Attention in Maritime Object Detection,” IEEE Trans. Neural Netw. Learn. Syst., vol. 35, no. 5, pp. 2410–2422, May 2024.

[8] X. Chen et al., “Spotlighting on Objects: Prior-Knowledge-Driven Maritime Image Dehazing and Object Detection Framework,” J. Mar. Sci. Eng., vol. 11, no. 4, pp. 339–351, Apr. 2024.

[9] A. M. Rekavandi, L. Xu, F. Boussaid, A.-K. Seghouane, S. Hoefs, dan M. Bennamoun, “A Guide to Image- and Video-Based Small Object Detection Using Deep Learning: Case Study of Maritime Surveillance,” IEEE Trans. Intell. Transp. Syst., vol. 26, no. 3, hlm. 2851–2869, Mar. 2025, doi:10.1109/TITS.2025.3530678.

[10] S. Tan et al., “DAShip: A Large-Scale Annotated Dataset for Ship Detection Using Distributed Acoustic Sensing,” Remote Sens., vol. 16, no. 2, p. 210, Feb. 2024.

[11] Y. Yuan et al., “AFF-LightNet: A Lightweight Ship Detection Architecture Based on Attentional Feature Fusion,” J. Mar. Sci. Eng., vol. 13, art. 44, Jan. 2025.

[12] Y. Wang et al., “YOLO-StarLS: A Ship Detection Algorithm Based on Wavelet Transform and Multi-Scale Feature Extraction for Complex Environments,” Symmetry, vol. 17, no. 7, art. 1116, Jul. 2025.

[13] K. Lan et al., “High-Efficiency and High-Precision Ship Detection Algorithm Based on Improved YOLOv8n,” Mathematics, vol. 12, no. 7, art. 1072, Apr. 2024.

[14] Y. Li, T. Yao, Y. Pan and T. Mei, "Contextual Transformer Networks for Visual Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 2, pp. 1489-1500, 1 Feb. 2023.

[15] J. Hu, L. Shen and G. Sun, "Squeeze-and-Excitation Networks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 7132-7141.

[16] W. Wu et al., “CGDU-DETR: An End-to-End Detection Model for Ship Detection in Day–Night Transition Environments,” J. Mar. Sci. Eng., vol. 13, no. 6, art. 1155, Jun. 2025.

[17] S. Sun et al., “Research on the Automatic Detection of Ship Targets Based on an Improved YOLO v5 Algorithm,” Mathematics, vol. 12, no. 11, art. 1714, Nov. 2024.

[18] Q. Qiu et al., “YOLOv7oSAR: A Lightweight High-Precision Ship Detection Model for SAR Images Based on the YOLOv7 Algorithm,” Remote Sens., vol. 17, no. 4, art. 835, Apr. 2025.

[19] R. D. Kwon, J. Kim, and T. Yoon, “Enhancing Ship Classification in Optical Satellite Imagery: Integrating CBAM with ResNet for Improved Performance,” arXiv preprint arXiv:2404.02135, Apr. 2024.

[20] V. S. Sanikommu et al., “Edge Computing for Maritime Ship-Port Detection Using YOLO,” Front. Artif. Intell., vol. 8, art. 1508664, Feb. 2025.

[21] Y. Zhou et al., “YOLOv7-Ship: A Lightweight Algorithm for Ship Object Detection in Complex Marine Environments,” J. Mar. Sci. Eng., vol. 13, no. 3, art. 389, Mar. 2025.

[22] N. Toprak and Y. Yalman, “Ship Detection from Optical Satellite Images Using CNN,” Turk. J. Eng., vol. 9, no. 2, pp. 88–97, Jun. 2025.

[23] B. E. Ayesha, T. Khan, and R. U. Khan, “Ship Detection in Remote Sensing Imagery for Arbitrarily Oriented Object Detection,” arXiv preprint arXiv:2503.14534, Mar. 2025.

[24] P. Lu et al., “LH-YOLO: A Lightweight and High-Precision Ship Detection Model Based on Improved YOLOv8n,” Remote Sens., vol. 16, no. 22, art. 4340, Nov. 2024.

[25] Y. Wu, Z. Chen, and W. Zhang, “Ship-YOLO: A Deep Learning Approach for Ship Detection in Remote Sensing Images,” arXiv preprint arXiv:2503.14534, Mar. 2025.

[26] S. Kang, Z. Hu, L. Liu, K. Zhang, and Z. Cao, ‘Object detection YOLO algorithms and their industrial applications: Overview and comparative analysis’, Electronics, vol. 14, no. 6, p. 1104, 2025.

[27] P. Zhao, ‘SPFFNet: Strip perception and feature fusion spatial pyramid pooling for fabric defect detection’, arXiv preprint arXiv:2502. 01445, 2025.

[28] J. Jing and C. Li, ‘Identification of lightweight rail defects based on YOLOv11 improvement’, in Fifth International Conference on Optical Imaging and Image Processing (ICOIP 2025), 2025, vol. 13688, pp. 512–517

[29] X. Chen, N. Jiang, Z. Yu, W. Qian, and T. Huang, ‘Citrus leaf disease detection based on improved YOLO11 with C3K2’, in International Conference on Computer Graphics, Artificial Intelligence, and Data Processing (ICCAID 2024), 2025, vol. 13560, pp. 746–751.

[30] J. Song, J. Xie, Q. Wang, and T. Shen, ‘An improved YOLO-based method with lightweight C3 modules for object detection in resource-constrained environments’, The Journal of Supercomputing, vol. 81, no. 5, p. 702, 2025

[31] Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., & Han, J. (2024). Yolov10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems*, *37*, 107984-108011.

[32] Tian, Y., Ye, Q., & Doermann, D. (2025). Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*.