

Original Article

e-ISSN: 2581-0545 - <https://journal.itera.ac.id/index.php/jsat/>

Received 00th January 20xx
Accepted 00th Febuary 20xx
Published 00th March 20xx

Open Access

DOI: 10.35472/x0xx0000

Deteksi dan Perhitungan Jumlah Sampah Pantai Menggunakan RT-DETR

Miftahul Huda^{*1}, Arsyiah Azahra², Putri Maulida Chairani³, Dimas Rizky Ramadhan⁴, Nabila Azhari⁵, Ade Lailani⁶

^{*} Program Studi Sains Data, Fakultas Sains, Institut Teknologi Sumatera

* Corresponding E-mail: miftahul.121450125@student.itera.ac.id

Abstract: DETR (*Detection Transformer*) is a Transformer-based object detection model that focuses on an end-to-end approach. However, due to the high computational complexity of the Transformer, the DETR model is not ideal for real-time inference. As a solution to the shortcomings of the DETR model, RT-DETR was developed to provide fast and efficient performance for real-time inference. This research aims to develop an automatic detection system to detect beach litter using RT-DETR. This system supports environmental conservation with accurate and real-time litter detection. Through this model, a performance comparison based on the number of parameters was conducted to assess the implementation of this technology in the field. The evaluation results show that RT-DETR Extra-Large is slightly superior in mAP@50 (0.816 vs. 0.810) and mAP@50-95 (0.612 vs. 0.606), while RT-DETR Large is faster in inference. This research provides a deep insight into the application of Transformer-based detection technology for the environment.

Keywords: RT-DETR, Real-time object detection, Litter on the Beach.

Abstrak: DETR (*Detection Transformer*) yaitu model deteksi objek berbasis *Transformer* yang berfokus pada pendekatan *end-to-end*. Namun, karena kompleksitas komputasi *Transformer* yang tinggi, model DETR kurang ideal untuk *inference* secara *real-time*. Sebagai solusi atas kekurangan dari model DETR tersebut, dikembangkanlah RT-DETR untuk memberikan performa yang cepat dan efisien untuk inferensi *real-time*. Penelitian ini bertujuan untuk mengembangkan sistem deteksi otomatis guna mendeteksi sampah di pantai menggunakan RT-DETR. Sistem ini mendukung konservasi lingkungan dengan pendekripsi sampah akurat dan *real-time*. Melalui model ini, perbandingan performa berdasarkan jumlah parameter dilakukan untuk menilai implementasi teknologi ini di lapangan. Hasil evaluasi menunjukkan bahwa RT-DETR Extra-Large sedikit lebih unggul dalam mAP@50 (0.816 vs 0.810) dan mAP@50-95 (0.612 vs 0.606), sementara RT-DETR Large lebih cepat dalam inferensi. Penelitian ini memberikan wawasan mendalam tentang aplikasi teknologi deteksi berbasis *Transformer* untuk lingkungan.

Kata Kunci : RT-DETR, Deteksi objek *real-time*, Sampah di Pantai.

Pendahuluan

Sampah pantai telah menjadi salah satu masalah lingkungan yang paling mendesak di seluruh dunia. Limbah yang menumpuk di sepanjang pantai bukan hanya mengganggu keindahan alam, tetapi juga memberikan dampak buruk bagi ekosistem laut. Plastik, yang merupakan salah satu jenis sampah paling umum, memakan waktu puluhan hingga ratusan tahun untuk terurai dan sering kali dikonsumsi oleh hewan laut secara tidak sengaja. Hal ini berdampak pada kesehatan ekosistem laut dan pada akhirnya dapat mempengaruhi manusia yang mengkonsumsi hasil laut tersebut [1]. Berdasarkan laporan dari World Bank, sekitar

60% dari total sampah laut adalah plastik, dan sebagian besar berasal dari aktivitas manusia di wilayah pesisir [2].

Seiring meningkatnya kesadaran akan pentingnya menjaga kebersihan laut, berbagai upaya telah dilakukan untuk mendekripsi dan membersihkan sampah di pantai. Salah satu pendekatan yang berkembang pesat adalah penggunaan teknologi berbasis kecerdasan buatan (Artificial Intelligence/AI). Teknologi ini memungkinkan deteksi sampah secara otomatis, yang sangat membantu dalam mempercepat proses identifikasi dan pengelolaan limbah. Pendekatan ini jauh lebih efisien dibandingkan dengan cara manual yang membutuhkan banyak tenaga dan waktu [3, 4].



RT-DETR (Real-Time Detection Transformer) merupakan salah satu model AI terbaru yang dirancang untuk mendeteksi objek secara real-time [8]. Model ini dirancang untuk meningkatkan akurasi deteksi tanpa mengorbankan kecepatan pemrosesan, yang menjadi kelemahan dari beberapa pendahulunya seperti DETR [5, 6]. Teknologi ini memanfaatkan arsitektur berbasis Transformer yang telah terbukti unggul dalam berbagai tugas deteksi objek, terutama pada dataset yang kompleks [8].

Melalui pelatihan menggunakan RT-DETR, diharapkan dapat dihasilkan model deteksi sampah pantai yang tidak hanya akurat tetapi juga efisien dalam hal waktu pemrosesan. Model ini dapat menjadi solusi potensial bagi berbagai pihak, mulai dari pemerintah hingga organisasi non-profit, dalam upaya membersihkan pantai dan menjaga ekosistem laut. Penggunaan teknologi ini juga membuka peluang untuk mengintegrasikan deteksi sampah dengan sistem pengelolaan limbah yang lebih modern, seperti drone atau robot pembersih pantai.

Penelitian ini bertujuan untuk mengembangkan sistem deteksi otomatis yang dapat membantu proses identifikasi dan pengumpulan sampah di pantai. Dengan menggunakan RT-DETR, penelitian ini juga berupaya untuk membandingkan performa model berdasarkan berbagai ukuran, seperti akurasi dan kecepatan, guna memastikan teknologi ini dapat diimplementasikan secara praktis di lapangan. Hal ini diharapkan dapat memberikan kontribusi nyata dalam mengatasi permasalahan sampah pantai yang selama ini menjadi tantangan global.

Metode

Dataset

Dataset yang digunakan dalam penelitian ini berasal dari *Roboflow*, yang dipublikasikan oleh Monash pada tahun 2024 dengan lisensi *Creative Commons Attribution 4.0 International* (CC BY 4.0) [10]. Dataset ini mencakup berbagai jenis material yang biasanya ditemukan sebagai limbah, yaitu 7 kategori objek sampah di pantai seperti “Bottle”, “Clothes”, “Metal”, “Plastic”, “Rope”, “Styrofoam”, dan “Wood”. Kami menggunakan 2675 gambar untuk pelatihan dan 841 gambar untuk validasi, namun dalam gambar validasi tidak memiliki kategori “wood” dan “clothes” [7]. Setiap gambar dalam dataset memiliki anotasi yang meliputi koordinat titik tengah, lebar, dan tinggi objek yang sudah dinormalisasi berdasarkan lebar dan tinggi gambar yang sebenarnya, serta kategori objek yang terdapat dalam koordinat tersebut.

Pelatihan Model RT-DETR

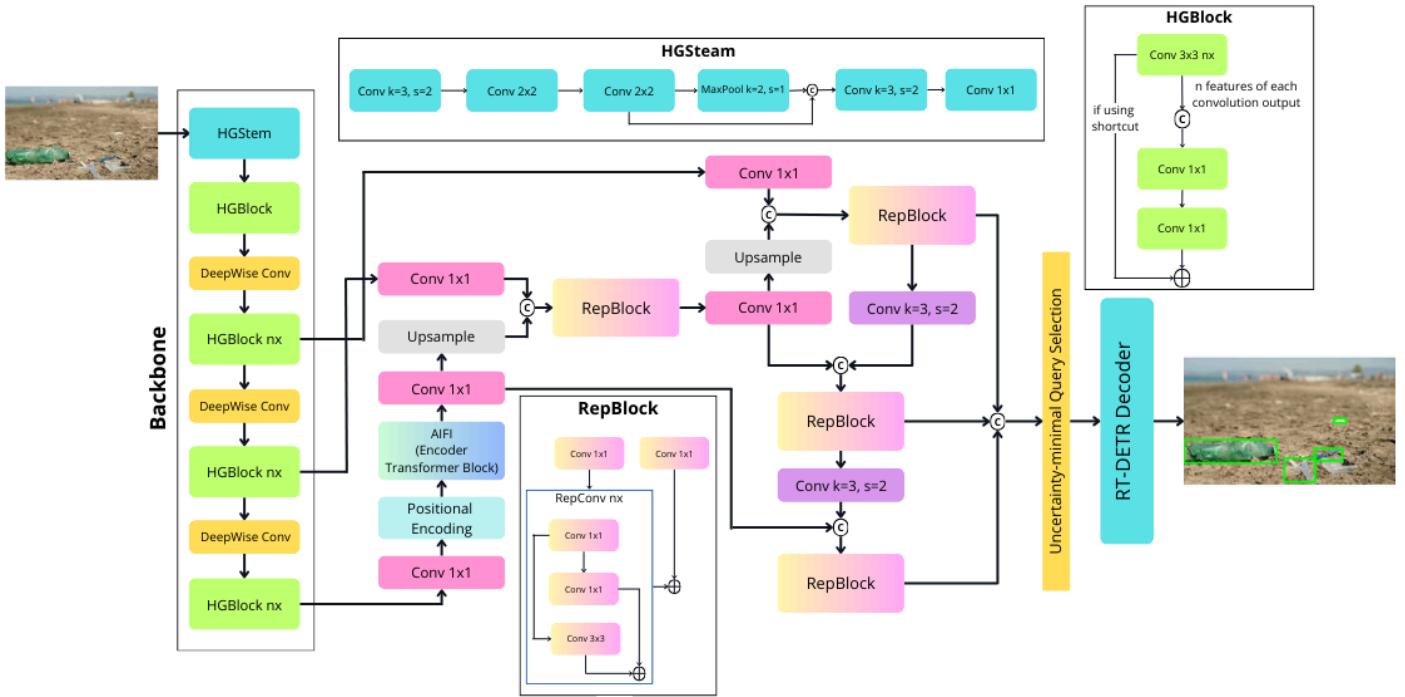
RT-DETR (*Real-Time Detection Transformer*) adalah model *real-time end-to-end* deteksi objek yang diusulkan oleh Zhao et al. (2023) dalam paper “*DETRs Beat YOLOs on Real-time Object Detection*” [8]. Sebelumnya DETR (*Detection Transformer*) yaitu model deteksi objek berbasis Transformer telah mendapat perhatian di kalangan peneliti, namun karena menggunakan biaya komputasi yang tinggi, kurang cocok untuk inferensi secara *real-time* [5, 8].

Kami melatih dua model *pre-trained* RT-DETR dengan *optimizer* adamW [11] menggunakan kode sumber dari repositori *Ultralytics*, yaitu model RT-DETR Large dan model RT-DETR Extra-Large untuk mengetahui apakah terdapat *trade-off* antara akurasi dan kecepatan berdasarkan jumlah parameter model pada dataset yang digunakan. Model RT-DETR Extra-Large tentunya memiliki jumlah parameter yang lebih banyak dan FLOPs yang lebih besar dibanding model RT-DETR Large, sehingga RT-DETR Extra-Large memiliki kecepatan yang lebih lambat dibanding RT-DETR Large.

Arsitektur RT-DETR

RT-DETR menggunakan CNN sebagai *backbone* untuk menghasilkan fitur multi skala dan menggunakan *Efficient Hybrid Encoder* yang dirancang untuk mempercepat pemrosesan fitur multi skala tersebut, sehingga cocok untuk inferensi *real-time* bahkan menyaingi kecepatan dari YOLOv8 [8, 9]. Selain itu, RT-DETR memilih fitur dari *encoder* dengan menerapkan *Uncertainty-minimal Query Selection* yang memilih fitur dengan ketidakpastian minimal dari perbedaan antara distribusi prediksi lokalisasi *P* dan klasifikasi *C* untuk meningkatkan akurasi, pemilihan objek kueri dioptimalkan dengan menambahkannya ke dalam *loss function* [8, 9]. Fitur-fitur terpilih dari *encoder* kemudian digunakan sebagai objek kueri untuk *decoder*.

Backbone yang digunakan model RT-DETR terdiri dari lapisan-lapisan HGSteam, HGBlock, dan Deep Wise Convolutional yang detailnya dapat dilihat pada **Gambar 2**. Perbedaan utama arsitektur RT-DETR Large dengan RT-DETR Extra-Large terletak pada jumlah HGBlock yang digunakan, di mana RT-DETR Large hanya menggunakan satu, tiga, dan satu HGBlock secara berurutan setelah Deep Wise Convolutional. Sedangkan model RT-DETR Extra-Large menggunakan dua, lima, dan dua HGBlock secara berurutan setelah Deep Wise Convolutional [9, 12]. HGBlock menggunakan *light convolutional* dengan kernel 3×3 sebanyak enam kali yang kemudian setiap output konvolusi digabung (*concatenate*) sehingga membentuk *skip connection* lalu melewati konvolusi 1×1 untuk menyesuaikan jumlah *channel*, namun HGBlock setelah HGSteam tidak menggunakan *light convolutional* melainkan



Gambar 2. Arsitektur RT-DETR. Fitur dengan tiga skala berbeda yang dihasilkan oleh backbone diproses melalui *Efficient Hybrid Encoder*. Fitur dari lapisan terakhir melewati lapisan Transformer, yang menerapkan mekanisme *Attention-based Intra-scale Feature Interaction* (AI FI) untuk melakukan *attention* antar channel. Ketiga fitur tersebut digabungkan, dan kemudian diterapkan *Uncertainty-minimal Query Selection* yang memilih fitur dari *encoder* sebagai objek kueri bagi *decoder*. Terakhir *decoder* dengan *heads* mengoptimalkan objek kueri untuk menghasilkan kategori objek dan koordinat *bounding boxes* objek.

konvolusi biasa pada umumnya [9]. Untuk masing-masing RepBlock menggunakan RepConv sebanyak tiga kali, RepConv menggabungkan (*Element-wise add*) konvolusi 3×3 dan 1×1 dari konvolusi 1×1 [8, 9]. HGSteam dan HGBlock menggunakan fungsi aktivasi ReLU sedangkan *Deep Wise Convolutional* menggunakan fungsi aktivasi SiLU [13] yang digunakan juga pada RepConv dan konvolusi setelah RepBlock dan AI FI [9].

Loss Function

RT-DETR memprediksi N objek atau sebanyak objek kueri dalam *decoder* [5]. Namun, jika N lebih besar dari jumlah objek dalam gambar, maka terdapat prediksi \emptyset atau prediksi kosong yang dianggap tidak ada objek, karena saat menghitung *loss function* jumlah prediksi harus sama dengan jumlah objek dalam gambar, sekaligus mengklasifikasikan antara objek dan bukan objek beserta kategorinya. Untuk mengatasi hal tersebut RT-DETR menggunakan *HungarianMatcher* untuk mencari pasangan optimal antara *bounding box* prediksi dan *bounding box ground truth* berdasarkan *loss function* tertentu [5, 9]:

$$\hat{id} = \arg \min_{id \in S_N} \sum_{i=1}^N L_{match}(y_i, \hat{y}_i) \quad (1)$$

pasangan optimal dicari menggunakan permutasi N elemen dengan nilai *loss* terendah yang didefinisikan sebagai [9, 14]:

$$L_{cls} = \alpha(1 - \hat{p})^\gamma (-\log(\hat{p})) - (1 - \alpha)\hat{p}^\gamma (-\log(1 - \hat{p})) \quad (2)$$

$$L_{bbox} = \sum_{i=1}^4 |b_i - \hat{b}_i| \quad (3)$$

$$L_{giou} = 1 - GIoU(\hat{b}_i, b_i) \quad (4)$$

$$L_{match} = \lambda_{cls} L_{cls} + \lambda_{bbox} L_{bbox} + \lambda_{giou} L_{giou} \quad (5)$$

dengan $\alpha = 0.25$, $\gamma = 2$, $\lambda_{cls} = 1$, $\lambda_{bbox} = 5$, $\lambda_{giou} = 2$ dan *GIoU* adalah Generalisasi *IoU* [16].

Setelah mendapat N pasangan yang optimal antara *bounding box* prediksi dan *bounding box ground truth* dengan $N = \sum \hat{id}$, maka *loss function* untuk pelatihan model RT-DETR dapat dihitung, yang didefinisikan sebagai [8, 9, 14-16]:

$$\begin{aligned}
 L = & \frac{\lambda_{giou}}{N} \sum_{i=1}^N GIoU(\hat{b}_i, b_i) + \frac{\lambda_{l1}}{N} \sum_{i=1}^N |\hat{b}_i - b_i| \\
 & + \frac{\lambda_{cls}}{N} \sum_{i=1}^N \sum_{c=1}^C [(q_{i,c} \log(\hat{p}_{i,c}) + (1 - q_{i,c}) \log(1 - \hat{p}_{i,c})) \\
 & \cdot (\alpha \hat{p}_{i,c}^{\gamma} (1 - p_{i,c}) + q_{i,c} p_{i,c})] \quad (5)
 \end{aligned}$$

dimana q adalah IoU antara *bounding box* prediksi dan *bounding box ground truth* dengan $\gamma = 2$ dan $\alpha = 0.25$ [9]. Kami menggunakan nilai *hyperparameter* λ bawaan dari repository *Ultralytics* yaitu $\lambda_{giou} = 2$, $\lambda_{l1} = 5$, dan $\lambda_{cls} = 1$ [9]. *Hyperparameter* λ digunakan sebagai penyeimbang atau bobot untuk masing-masing nilai *loss*.

Hasil dan Pembahasan

Kami melatih dan mengevaluasi kedua model RT-DETR menggunakan satu GPU P100 dengan *batch size* sebesar 12, kami membandingkan kompleksitas model berdasarkan jumlah parameter, GFLOPs (*Giga Floating Point Operations*) [17], serta kecepatan waktu inferensi.

Table 1. Jumlah parameter, waktu inferensi, dan GFLOPs pada model

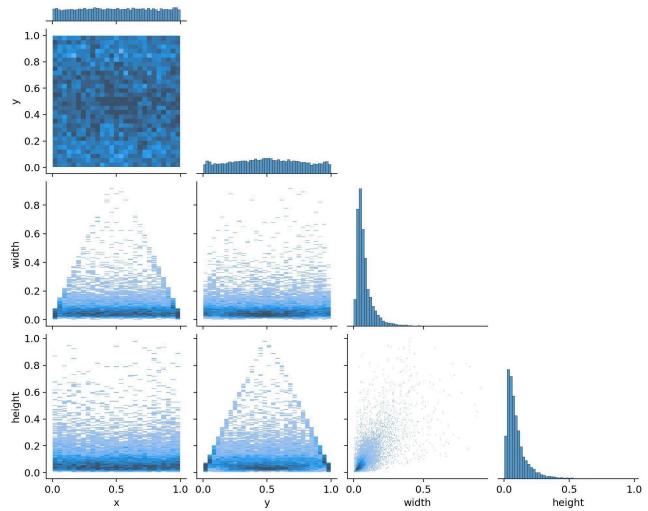
| Models | Params | GFLOPs | Inference |
|---------------------|--------|--------|-----------|
| RT-DETR Large | 32.9 M | 108 | 20.1 ms |
| RT-DETR Extra-Large | 67.3 M | 234.4 | 34.5 ms |

GFLOPs atau setara dengan 1 miliar FLOPs (*Floating Point Operations Per Second*) yang digunakan oleh model RT-DETR Extra-Large jauh lebih besar dibandingkan pada model RT-DETR Large. Hal ini menunjukkan bahwa model Extra-Large memerlukan lebih banyak komputasi setiap operasi, yang berdampak langsung pada penggunaan sumber daya dan waktu inferensi. Konsekuensinya, model RT-DETR Extra-Large memiliki tuntutan yang lebih besar terhadap perangkat keras, baik dari segi kapasitas memori maupun daya pemrosesan dan mungkin kurang efisien untuk aplikasi dengan keterbatasan sumber daya. Namun tidak menjadi masalah untuk sumber daya komputasi yang kuat seperti penggunaan GPU atau TPU.

Karakteristik Bentuk Objek

Umumnya dalam model-model *object detection* karakteristik objek terutama ukuran objek dapat mempengaruhi kemampuan model dalam mendeteksi objek, seperti pada YOLOv1 [18] oleh Redmon et al. yang dikenal karena kecepatannya, namun sulit dalam memprediksi objek kecil dari fitur map 7×7 . Kemudian Single Shot MultiBox Detector [19] oleh Liu et al. muncul dengan menggunakan multi skala fitur dengan ukuran yang

berbeda-beda dari berbagai lapisan dalam satu CNN yang digunakan untuk mendeteksi objek, tetapi fitur dari lapisan yang lebih dangkal tidak cukup untuk menangkap detail objek dengan baik. *Feature Pyramid Networks* [20] oleh Lin et al. muncul untuk mengatasi masalah ini dengan piramida fitur dari lapisan dalam menggunakan *upsample*, lalu fitur piramida digabung dengan fitur dari lapisan dangkal. Sehingga fitur dengan resolusi tinggi lebih detail dan berguna untuk memprediksi berbagai skenario objek termasuk objek yang kecil. Konsep-konsep tersebut menjadi dasar kebanyakan model *object detection* modern saat ini seperti versi-versi terbaru dari YOLO dan RT-DETR [8, 21-23].



Gambar 3. Distribusi dan hubungan dari koordinat, lebar, dan tinggi objek dalam dataset.

Berdasarkan **Gambar 3**, terlihat bahwa objek pada dataset cenderung sering muncul di bagian tengah gambar, seperti yang ditunjukkan oleh hubungan dan distribusi koordinat (x, y). Jika dilihat dari distribusi panjang dan lebar, banyak objek yang memiliki ukuran kecil, hal ini umum dalam merepresentasikan ukuran sampah-sampah di pantai pada gambar visual saat melakukan inferensi model. Meskipun kondisi tersebut memberikan representasi yang realistik terhadap kondisi sebenarnya, tetapi keberadaan objek kecil bisa menjadi tantangan bagi model *object detection* dalam mendeteksi dan mengklasifikasikan jenis-jenis sampah secara akurat.

Ukuran Performa dan Diagnosis Model

Seperti pada umumnya dalam mengevaluasi model deteksi objek mAP (*mean Average Precision*) digunakan sebagai metrik utama dalam mengevaluasi model, kami membandingkan evaluasi kedua model menggunakan mAP dengan *threshold confidence IoU* = 0.5 dan $IoU = 0.5 - 0.95$ serta dengan presisi dan recall dari

masing-masing kelas pada kedua model seperti yang diperlihatkan pada **Tabel 2** dan **Tabel 3**.

Kami menemukan perbedaan performa yang relatif kecil dan kurang signifikan antara kedua model. Meskipun terdapat *trade-off* terhadap penggunaan HGBlock yang lebih banyak, tetapi masih terdapat kategori kelas yang mengalami penurunan presisi, yaitu pada kategori kelas Styrofoam, dan penurun nilai recall pada kategori kelas Rope. Berdasarkan hasil evaluasi pada dataset yang kami gunakan, model Extra-Large menunjukkan peningkatan performa keseluruhan dibandingkan dengan model Large, terutama dalam hal klasifikasi dan identifikasi objek. Namun, recall yang sedikit lebih rendah pada model Extra-Large menunjukkan bahwa model ini menghasilkan sedikit lebih banyak *False Negatives* (FN) dibandingkan model Large. Selain itu, kecepatan inferensi RT-DETR Extra-Large lebih lambat dibandingkan RT-DETR Large, seperti yang ditunjukkan pada **Tabel 1**.

Table 2. Evaluasi precision, recall dan, mAP model RT-DETR Large.

| Class | Instances | Precision | Recall | mAP@50 | mAP@50-95 |
|-----------|-----------|-----------|--------|--------|-----------|
| Bottle | 874 | 0.906 | 0.814 | 0.872 | 0.638 |
| Metal | 458 | 0.882 | 0.852 | 0.878 | 0.648 |
| Plastic | 603 | 0.838 | 0.776 | 0.819 | 0.636 |
| Rope | 205 | 0.709 | 0.512 | 0.619 | 0.409 |
| Styrofoam | 418 | 0.904 | 0.765 | 0.861 | 0.697 |

Table 3. Evaluasi precision, recall dan, mAP model RT-DETR Extra-Large.

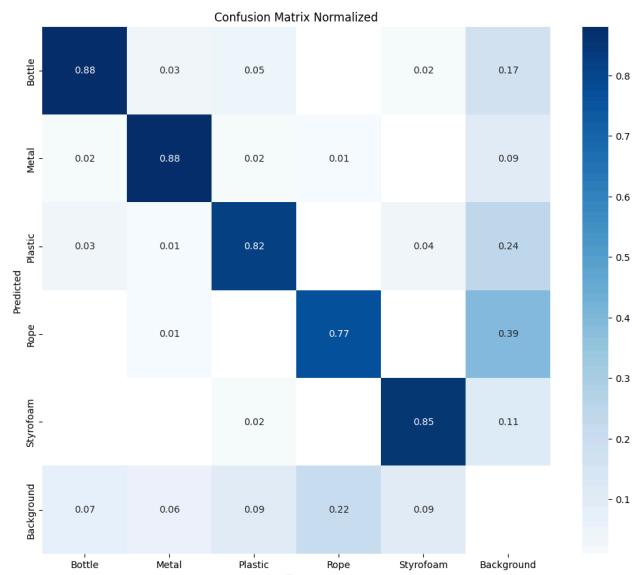
| Class | Instances | Precision | Recall | mAP@50 | mAP@50-95 |
|-----------|-----------|-----------|--------|--------|-----------|
| Bottle | 874 | 0.908 | 0.827 | 0.871 | 0.645 |
| Metal | 458 | 0.925 | 0.832 | 0.882 | 0.658 |
| Plastic | 603 | 0.847 | 0.774 | 0.813 | 0.637 |
| Rope | 205 | 0.796 | 0.478 | 0.64 | 0.413 |
| Styrofoam | 418 | 0.863 | 0.801 | 0.871 | 0.705 |

Table 4. Perbandingan evaluasi model Large dan model Extra-Large.

| Model | Precision | Recall | mAP@50 | mAP@50-95 |
|---------------------|-----------|--------|--------|-----------|
| RT-DETR Large | 0.848 | 0.744 | 0.810 | 0.606 |
| RT-DETR Extra-Large | 0.868 | 0.743 | 0.816 | 0.612 |

Kedua model memiliki nilai evaluasi mAP yang kurang baik pada kategori kelas Rope. Berdasarkan matriks konfusi pada **Gambar 4** dan **Gambar 7**, terlihat bahwa model sering mengalami kesalahan dalam memprediksi kategori kelas Rope dan Background. Model cenderung salah

mengklasifikasikan objek non-Rope sebagai Rope dan sebaliknya, yang mengakibatkan penurunan nilai presisi dan recall pada kategori tersebut. Kemudian kategori Plastic pada model Large dan kategori Bottle pada model Extra-Large masih cenderung menganggap non-Objek sebagai Objek. Keseluruhan kedua model menunjukkan performa *mean Average Precision* yang baik pada rata-rata ambang batas *IoU* 0.5 dan *IoU* 0.5–0.95 berdasarkan aturan evaluasi COCO [cite] pada dataset yang kami gunakan, meskipun dataset yang digunakan mengandung banyak objek kecil, seperti yang ditampilkan pada **Gambar 2**. Hal ini mengindikasikan bahwa model *object detection* modern seperti RT-DETR yang menggunakan konsep *Feature Pyramid Networks* dengan tiga skala fitur dan *Efficient Hybrid Encoder* memiliki kemampuan yang cukup baik, dengan menggabungkan informasi multi-skala untuk memperkuat representasi fitur berbagai ukuran objek.



Gambar 4. Confusion matrix evaluasi model RT-DETR Large.

Sehingga pendekatan ini membuat model mampu dalam mendeteksi objek berukuran kecil seperti sampah di pantai. **Generalisasi Model**

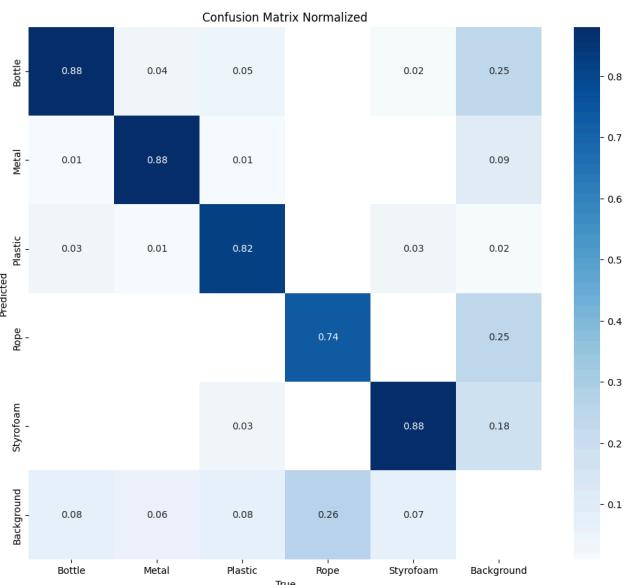
Kami melakukan pengujian generalisasi model Large dan Extra-Large dengan ambang batas *confidence score* 0.2 terhadap tiga kondisi umum pada pantai dengan data random, yaitu: (1) ketika terdapat hanya beberapa sampah, (2) ketika terdapat banyak sampah, dan (3) ketika tidak terdapat sampah di pantai. Hasil ditunjukkan pada **Gambar 5** dan **Gambar 6**. Pada gambar pertama, model Large mendeteksi dua objek sampah yaitu botol plastik dan



Gambar 5. Prediksi objek menggunakan model RT-DETR Large.



Gambar 6. Prediksi objek menggunakan model RT-DETR Extra-Large.



Gambar 7. Confusion matrix evaluasi model RT-DETR Extra-Large.

plastik kecil pada area pasir yang terlihat lebih bersih dengan hanya beberapa objek sampah, model cukup akurat karena objek yang terdeteksi memang menyerupai botol

plastik dan plastik kecil. Sedangkan model Extra-Large mendeteksi lebih banyak objek yang sebelumnya tidak terdeteksi (styrofoam dan lebih banyak plastik), perbedaan ini menunjukkan model Extra-Large mungkin lebih sensitif dalam mengenali berbagai jenis sampah.

Pada gambar kedua, model Large mendeteksi 42 objek sampah pada area pantai yang terlihat sangat kotor dengan banyak jenis sampah tersebar, model terlihat mampu mengidentifikasi berbagai jenis sampah dengan baik, meskipun mungkin terdapat beberapa *False Positives*. Untuk model Extra-Large lebih konservatif dalam deteksi, kemungkinan untuk mengurangi *False Positives*. Namun, ini dapat menyebabkan beberapa sampah nyata tidak terdeteksi, seperti pada gambar kedua, di mana jumlah sampah yang terdeteksi lebih sedikit dibandingkan dengan hasil model Large.

Pada gambar ketiga, tidak ada sampah yang terdeteksi pada area pantai bersih dengan pasir dan laut tanpa ada tanda-tanda keberadaan sampah. Dalam kasus ini kinerja kedua model tampaknya baik, karena memang area tersebut terlihat bersih tanpa sampah.

Kesimpulan

Hasil penelitian menunjukkan bahwa model RT-DETR Large lebih baik berdasarkan GFLOPs dan waktu kecepatan inferensi untuk mengidentifikasi sampah di pantai dibandingkan model RT-DETR Extra-Large. Namun, hasil evaluasi menunjukkan bahwa model RT-DETR Extra-Large memiliki kinerja sedikit lebih baik dalam hal keseluruhan metrik yaitu presisi, recall, dan mAP, meskipun masih terdapat pertimbangan antara akurasi dan kecepatan inferensi. Penelitian ini memberikan kontribusi yang signifikan untuk konservasi lingkungan dan pengelolaan sampah, tetapi model ini masih menghadapi masalah dalam menemukan objek kecil dan kesalahan klasifikasi, terutama dalam kategori "Rope". Pengembangan metode yang lebih canggih untuk meningkatkan akurasi deteksi objek kecil dan optimalisasi kecepatan inferensi tanpa mengurangi akurasi adalah saran untuk penelitian lanjutan.

Konflik Kepentingan. Penulis menyatakan tidak ada konflik kepentingan yang perlu dideklarasikan.

Ucapan Terima Kasih. Pekerjaan ini didukung sebagian besar oleh Program Studi Sains Data Institut Teknologi Sumatera. Ucapan terima kasih kepada Ade Lailani dan Christyan Tamara Nadeak atas saran yang bermanfaat tentang penulisan, presentasi, dan pembahasan.

Referensi

- [1] World Bank, Plastic Waste Management in Coastal Areas, 2022.
- [2] J. Jambeck *et al.*, "Plastic waste inputs from land into the ocean," *Science*, vol. 347, no. 6223, pp. 768-771, 2015.
- [3] Y. Liu *et al.*, "Efficient Multi-scale Feature Fusion for Object Detection," *IEEE Access*, vol. 9, pp. 24539-24550, 2021.
- [4] D. Borthakur and A. Singh, "Waste management in coastal areas: Challenges and solutions," *Marine Pollution Bulletin*, vol. 150, pp. 110-118, 2019.
- [5] N. N. Carion *et al.*, "End-to-End Object Detection with Transformers," in *Proc. ECCV*, 2020, pp. 213-229.
- [6] C. He *et al.*, "Sparse R-CNN: End-to-End Object Detection with Learnable Proposals," *CVPR*, 2021, pp. 14454-14463.
- [7] Monash, "Beach Waste Dataset," Roboflow Universe. Roboflow, Oct. 2024. [Online]. Available: <https://universe.roboflow.com/monash-ventz/beach-waste-vqths>. Accessed: Nov. 19, 2024.
- [8] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "DETRs Beat YOLOs on Real-time Object Detection," *arXiv preprint*, 2024. [Online]. Available: <https://arxiv.org/abs/2304.08069>.
- [9] Ultralytics, "Ultralytics Neural Network Modules," GitHub repository, 2024. [Online]. Available: <https://github.com/ultralytics/ultralytics/tree/main/ultralytics/nn/modules>. [Accessed: Dec. 1, 2024].
- [10] Creative Commons, "Attribution 4.0 International (CC BY 4.0) License," [Online]. Available: <https://creativecommons.org/licenses/by/4.0/>. [Accessed: Dec. 1, 2024].
- [11] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," *arXiv*, 2019. [Online]. Available: <https://arxiv.org/abs/1711.05101>.
- [12] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *arXiv*, 2017. [Online]. Available: <https://arxiv.org/abs/1610.02357>.
- [13] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning," *arXiv*, 2017. [Online]. Available: <https://arxiv.org/abs/1702.03118>.
- [14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," *arXiv*, 2018. [Online]. Available: <https://arxiv.org/abs/1708.02002>.
- [15] H. Zhang, Y. Wang, F. Dayoub, and N. Sünderhauf, "VarifocalNet: An IoU-aware Dense Object Detector," *arXiv*, 2021. [Online]. Available: <https://arxiv.org/abs/2008.13367>.
- [16] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression," *arXiv*, 2019. [Online]. Available: <https://arxiv.org/abs/1902.09630>.
- [17] R. Desislavov, F. Martínez-Plumed, and J. Hernández-Orallo, "Trends in AI inference energy consumption: Beyond the performance-vs-parameter laws of deep learning," *Sustainable Computing: Informatics and Systems*, vol. 38, p. 100857, Apr. 2023. [Online]. Available: <https://arxiv.org/abs/2109.05472>.
- [18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *arXiv*, 2016. [Online]. Available: <https://arxiv.org/abs/1506.02640>.
- [19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *Computer Vision – ECCV 2016*, Springer, 2016, pp. 21–37. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46448-0_2.
- [20] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," *arXiv*, 2017. [Online]. Available: <https://arxiv.org/abs/1612.03144>.
- [21] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-Time Flying Object Detection with YOLOv8," *arXiv*, 2024. [Online]. Available: <https://arxiv.org/abs/2305.09972>.
- [22] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "YOLOv10: Real-Time End-to-End Object Detection," *arXiv*, 2024. [Online]. Available: <https://arxiv.org/abs/2405.14458>.
- [23] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," *arXiv*, 2024. [Online]. Available: <https://arxiv.org/abs/2410.17725>.