



## Simulasi Perbandingan Training Dataset YOLOv5 dan YOLOv8 Untuk Deteksi Objek Bawah Air Pada Prototipe AUV

Ryan Yudha Aditya<sup>1</sup>, Erutan Dwi Sajiwo<sup>2</sup>, Zindhu Maulana Ahmad Putra<sup>3</sup>, M. Yusuf Santoso<sup>4</sup>, M. Khoirul Hasin<sup>5</sup>, Adianto<sup>6</sup>, Dimas Pristovani Riananda<sup>7</sup>, Isa Rachman<sup>8</sup>, Agus Khumaidi<sup>9</sup>, Yuning Widiarti<sup>10</sup>

<sup>1, 2, 5, 8, 9</sup>Program Studi Teknik Otomasi, Jurusan Teknik Kelistrikan Kapal, Politeknik Perkapalan Negeri Surabaya Kampus ITS Sukolilo Jl. Teknik Kimia, Keputih Sukolilo, Surabaya, Indonesia

<sup>4</sup>Program Studi Teknik Keselamatan dan Kesehatan Kerja, Jurusan Teknik Permesinan Kapal, Politeknik Perkapalan Negeri Surabaya Kampus ITS Sukolilo Jl. Teknik Kimia, Keputih Sukolilo, Surabaya, Indonesia

<sup>3, 6, 7, 10</sup>Program Studi Teknik Kelistrikan Kapal, Jurusan Teknik Kelistrikan Kapal Politeknik Perkapalan Negeri Surabaya Kampus ITS Sukolilo Jl. Teknik Kimia, Keputih Sukolilo, Surabaya, Indonesia

<sup>1</sup>[ryanyudhaaditya@ppns.ac.id](mailto:ryanyudhaaditya@ppns.ac.id), <sup>2</sup>[erutandwi@student.ppns.ac.id](mailto:erutandwi@student.ppns.ac.id), <sup>3</sup>[zindhu@ppns.ac.id](mailto:zindhu@ppns.ac.id), <sup>4</sup>[yusuf.santoso@ppns.ac.id](mailto:yusuf.santoso@ppns.ac.id),  
<sup>5</sup>[khoirul.hasin@ppns.ac.id](mailto:khoirul.hasin@ppns.ac.id), <sup>6</sup>[Adianto@ppns.ac.id](mailto:Adianto@ppns.ac.id), <sup>7</sup>[dimaspristovani@ppns.ac.id](mailto:dimaspristovani@ppns.ac.id), <sup>8</sup>[isarachman@ppns.ac.id](mailto:isarachman@ppns.ac.id),  
<sup>9</sup>[aguskhumaidi@ppns.ac.id](mailto:aguskhumaidi@ppns.ac.id), <sup>10</sup>[yuning.widiarti@ppns.ac.id](mailto:yuning.widiarti@ppns.ac.id)

### Abstract

This study presents an evaluation of the training performance of YOLOv5n and YOLOv8n models on image datasets to optimize real-time object detection on an autonomous underwater vehicle (AUV) prototype. The dataset consists of 4,114 images with three object classes: gates, obstacles, and drums. The training process uses the Google Colab platform with 200 epochs, a batch size of 16, and an image resolution of 640x640 pixels. The results show that YOLOv8n outperforms YOLOv5n for the precision parameter of 0.994 and the mAP50-95 parameter of 0.940. YOLOv8n is proven to be more effective for underwater object detection on the AUV prototype, supporting more accurate and efficient navigation and environmental recognition.

Keywords: YOLOv5n, YOLOv8n, object detection, AUV

### Abstrak

Penelitian ini menyajikan evaluasi unjuk kerja pelatihan model YOLOv5n dan YOLOv8n pada dataset citra untuk mengoptimalkan deteksi objek real-time pada prototipe kendaraan bawah air otonom (AUV). Dataset yang dimaksud terdiri dari 4.114 gambar dengan tiga kelas objek, yaitu objek gerbang, rintangan, dan drum. Proses pelatihan menggunakan platform Google Colab dengan 200 epoch, ukuran batch 16, dan resolusi gambar 640x640 piksel. Hasil menunjukkan bahwa YOLOv8n melampaui YOLOv5n untuk parameter presisi yaitu 0,994 dan parameter mAP50-95 yaitu 0,940. YOLOv8n terbukti lebih efektif untuk deteksi objek bawah air pada prototipe AUV, mendukung navigasi dan pengenalan lingkungan lebih akurat dan efisien.

Kata kunci: YOLOv5n, YOLOv8n, deteksi objek, AUV

Diterima Redaksi : 18-05-2025 | Selesai Revisi : 21-05-2026 | Diterbitkan Online : 22-06-2026

### 1. Pendahuluan

Pertumbuhan teknologi mengalami kemajuan yang signifikan, terutama pada bidang robotika [1]. Robotika banyak diterapkan dalam berbagai aplikasi, salah satunya adalah pada aplikasi pemetaan bawah air (batimetri) dengan menggunakan *Remotely Operated Vehicle* (ROV) dan *Autonomous Underwater Vehicle* (AUV) [2]. Terdapat perbedaan pada kedua robot tersebut, yaitu ROV yang menggunakan tambatan kabel sebagai sarana komunikasi dan kendali manuver perangkat agar bisa beroperasi bawah air, sedangkan AUV dapat beroperasi secara otonom [3] [4].

Pada Penggunaan AUV, deteksi objek secara *real-time* telah menjadi fungsi utama dalam mendukung tugas operasional AUV, sebagai dasar kondisi untuk melakukan manuver, pemantauan objek bawah air, hingga rekonstruksi permukaan bawah air (*volumetric reconstruction*) [5]. Dengan keterbatasan visual dan gangguan bawah air yang dinamis, sehingga dibutuhkan skema pengolahan citra dengan fitur konvolusi yang mampu mengenali detail piksel objek secara akurat. Dimana salah satu metode yang umum digunakan adalah algoritma You Only Look Once (YOLO). Algoritma YOLO merupakan algoritma yang pertama kali diperkenalkan sebagai metode untuk

deteksi objek *real-time* [6]. Dalam YOLO, proses deteksi dilakukan dengan teknik yang memiliki komputasi ringan dan efisien, yang menggabungkan *bounding box* dan kemungkinan kelas objek dalam satu sistem [7].

Perkembangan variasi YOLO, seperti YOLOv5 dan YOLOv8 memiliki optimalisasi kecepatan dan akurasi yang representatif [8]. YOLOv5 mengandalkan struktur yang menggunakan *anchor* untuk mendeteksi objek dengan cara yang efektif, sementara YOLOv8 memperkenalkan metode tanpa *anchor* yang lebih sederhana dan efisien [9]. YOLOv5 dan YOLOv8 telah terbukti efektif dalam aplikasi tertentu secara *real-time* [8] [10].

Pada penelitian sebelumnya, YOLOv4 diterapkan pada bidang *Unmanned Surface Vehicle* (USV) dengan jumlah 954 dataset [11] dan pada deteksi kecacatan hasil pengelasan dengan jumlah 1000 dataset [12]. Untuk YOLOv5, pernah digunakan untuk identifikasi warna buoy dengan metode pada miniatur *Autonomous Surface Vehicle* (ASV) dengan jumlah 1200 dataset [13], pada sistem deteksi api pada graving dock dengan jumlah 2745 dataset [14]. Dalam konteks kompetisi robot, YOLOv5 digunakan untuk peningkatan akurasi deteksi bola dengan jumlah 2611 dataset [15].

Penelitian ini bertujuan untuk memperoleh algoritma yang optimal, dengan jalan membandingkan hasil Training YOLOv5n dan YOLOv8n pada dataset citra yang digunakan dalam Kompetisi Robot Berbasis Artificial Intelligence (KRBAI) 2024. Dalam konteks robotika, deteksi objek digunakan untuk tugas seperti navigasi dan pengenalan lingkungan, seperti yang diterapkan pada robot bawah air [16]. Lebih lanjut, penelitian ini berkontribusi dalam observasi arsitektur deteksi objek yang optimal baik pada YOLOv5n atau YOLOv8n. Temuan dari studi perbandingan ini akan menjadi dasar bagi pengembangan sistem persepsi AUV di masa mendatang, khususnya dalam memilih model pembelajaran mendalam yang optimal untuk deteksi objek secara *real-time* dalam skenario bawah air yang dinamis, sehingga meningkatkan kemampuan navigasi dan efektivitas operasional.

## 2. Metode Penelitian

### 2.1. Underwater Robot

*Underwater robot* yang digunakan pada penelitian ini merupakan robot yang beroperasi di bawah air menggunakan *thruster* sebagai penggerak dan bermanuver [17] memanfaatkan 6 *Degrees of Freedom* (6 DOF) [4]. Gambar 1 menunjukkan bentuk *prototype robot underwater*, dimana rangka menggunakan *High Density Polyethylene* (HDPE) dengan ketebalan 10 mm. Penggunaan HDPE dipilih karena efisien, mudah dalam perawatannya, dan memiliki ketahanan terhadap korosi [18]. Robot ini memiliki dimensi 65 x 44 x 28 cm (PxLxT) dan memiliki berat skitar 19 kg. Robot ini digunakan untuk sarana pengambilan data pada lintasan yang ada di lapangan.



Gambar 1. Bentuk Prototipe Robot Underwater

### 2.2. Pengumpulan Dataset Citra

Pengumpulan dataset merupakan suatu komponen terpenting dan digunakan untuk pengenalan objek [19]. Pada penelitian ini dataset yang digunakan memiliki tiga klasifikasi bentuk yaitu *gate*, *obstacle*, dan *drum*. Contoh dataset yang digunakan dapat dilihat pada Tabel 1. Total jumlah dataset untuk penelitian ini adalah 4114 data citra, dengan menggunakan tiga bentuk objek yang berbeda.

Tabel 1. Dataset Penelitian

| Jenis Objek | Gambar Dataset                                                                        | Jumlah | Total |
|-------------|---------------------------------------------------------------------------------------|--------|-------|
| Gate        |    | 1881   | 4114  |
| Obstacle    |   | 1112   |       |
| Drum        |  | 1909   |       |

### 2.3. Labelling Dataset

*Labelling* merupakan tahapan proses, dimana setiap gambar harus diberikan label sesuai dengan *class*-nya masing masing. Proses pelabelan dilakukan dengan cara memberikan kotak atau *bounding box* pada gambar yang akan dideteksi. Pada penelitian ini, *gate* termasuk dalam *class 0*, *obstacle* termasuk dalam *class 1*, dan *drum* termasuk dalam *class 2*. Hasil proses labelling gambar berupa anotasi yang menampilkan empat koordinat dengan nama *x\_center* (X1), *y\_center* (Y1), *width*(X2), dan *Height* (Y2) sebagai dasar pembuatan *bounding box*. Contoh anotasi gambar setelah proses labelling dapat dilihat pada Tabel 2.

Tabel 2. Anotasi Dataset

| Class | X1    | Y1    | X2    | Y2    |
|-------|-------|-------|-------|-------|
| 0     | 0.273 | 0.276 | 0.240 | 0.198 |
| 0     | 0.274 | 0.272 | 0.2   | 0.188 |
| 1     | 0.486 | 0.253 | 0.122 | 0.502 |
| 1     | 0.488 | 0.254 | 0.126 | 0.501 |
| 2     | 0.579 | 0.372 | 0.101 | 0.069 |
| 2     | 0.579 | 0.370 | 0.101 | 0.073 |

Tabel 2 menyajikan bentuk hasil koordinat *YOLOv5* dapat dilihat pada Gambar 2 dan *YOLOv8* dapat dilihat pada Gambar 3. *bounding box* dan *class* dari masing masing objek. Dataset yang telah dianotasikan akan diolah menjadi *ground-truth* dan dibandingkan antara aktual dengan *predicted box* serta dievaluasi menggunakan *confusion matrix*.

## 2.4. Konfigurasi Training Dataset

Konfigurasi diperlukan untuk mengatur dataset yang akan ditraining pada platform Google Collabs. Konfigurasi *training* dataset yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3. Dari Tabel 3, dapat diketahui nilai iterasi per *Epoch*. Observasi *Epoch* digunakan untuk menghasilkan pelatihan dataset yang optimal [20].

Tabel 3. Konfigurasi Training Dataset

| Jenis Konfigurasi | YOLOv5  | YOLOv8  |
|-------------------|---------|---------|
| Batch             | 16      | 16      |
| Epoch             | 200     | 200     |
| Image Size        | 640     | 640     |
| Model             | Yolov5n | Yolov8n |
| Ukuran            | 3.7 MB  | 6.3 MB  |

## 2.5. Training Dataset

*Training Dataset* atau tahapan pelatihan, berfungsi untuk melatih model dalam mengenali objek yang diinginkan [21]. Pada *training* dataset ini, digunakan *Graphics Processing Unit* (GPU) yang dinilai lebih cepat dibanding menggunakan *Central Processing Unit* (CPU) [22]. Penelitian ini membandingkan *Precision*, *Recall*, *Mean Average Precision 50* (mAP), dan mAP 50-95, sebagai parameter unjuk kerja performa algoritma yang dibandingkan. Perhitungan manual *Precision* dapat dilihat pada Persamaan 1.

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

Perhitungan manual *Recall* dapat dilihat pada Persamaan 2.

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

Perhitungan manual *Mean Average Precision 50* (mAP50) dapat dilihat pada Persamaan 3.

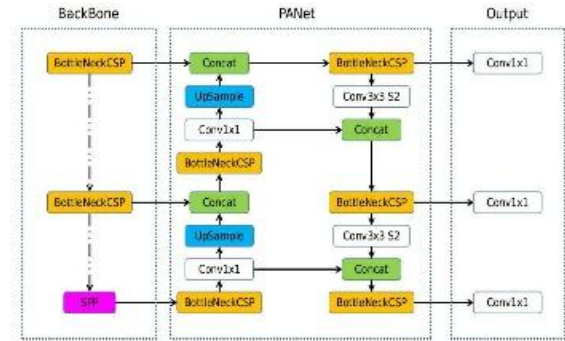
$$mAP50 = \frac{1}{N} \sum_{i=1}^N AP_i(IoU = 0.5) \quad (3)$$

Perhitungan manual mAP50-95 dapat dilihat pada Persamaan 4.

$$mAP50 - 95 = \frac{1}{N \cdot 10} \sum_{i=1}^N \sum_{t=0.5}^{0.95, step=0.05} AP_i(t) \quad (4)$$

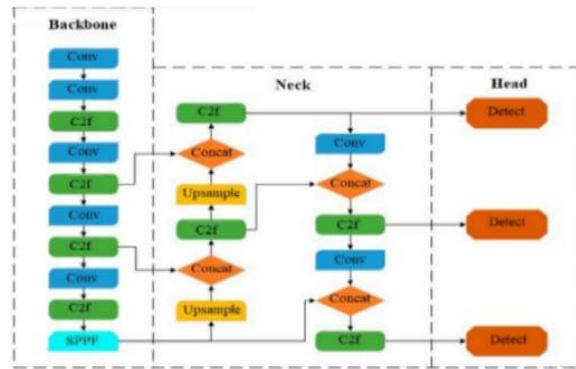
## 2.6. Arsitektur YOLO

Arsitektur YOLO merupakan suatu rancangan model yang terdiri dari tiga bagian utama yaitu Backbone, Neck, dan Head [23]. Bentuk arsitektur



Gambar 2. Arsitektur YOLOv5

Pada Gambar 2, YOLOv5 menggunakan modul *BottleNeckCSP* untuk mengekstrak fitur, menggunakan head berbasis *anchor box* [24], dan PANet untuk menggabungkan fitur dari berbagai skala dengan operasi *Concat* dan *UpSample*.



Gambar 3. Arsitektur YOLOv8

Pada Gambar 3, YOLOv8 mengganti modul C2f dengan C3, sehingga dapat meningkatkan performa dengan mekanisme penghubung cabang yang lebih banyak untuk aliran gradien yang lebih kompleks, sekaligus menjaga model tetap ringan [25].

## 3. Hasil dan Pembahasan

### 3.1 Hasil Training Dataset

Setelah dilakukan proses *training* dataset menggunakan dua model berbeda, dapat diketahui proses waktu proses training sekitar 3 jam 31 menit pada YOLOv5 dan 3 jam 17 menit pada YOLOv8. Tampilan setelah proses training dataset YOLOv5 dan YOLOv8 dapat dilihat pada Gambar 4 dan Gambar 5.

```

with torch.cuda.amp.autocast(enabled=True):
    199/199   2.13s   0.01252   0.005697   0.001264   45   640: 98% 254/258 [00:31:00:00, 8.1011/s]/c
with torch.cuda.amp.autocast(enabled=True):
    199/199   2.13s   0.01251   0.005697   0.001261   44   640: 95% 255/258 [00:31:00:00, 7.8711/s]/c
with torch.cuda.amp.autocast(enabled=True):
    199/199   2.13s   0.01253   0.005694   0.001271   40   640: 95% 256/258 [00:32:00:00, 8.0111/s]/c
with torch.cuda.amp.autocast(enabled=True):
    199/199   2.13s   0.01253   0.005694   0.00127   48   640: 100% 257/258 [00:32:00:00, 8.0311/s]/c
with torch.cuda.amp.autocast(enabled=True):
    199/199   2.13s   0.01253   0.005692   0.001266   5   640: 100% 258/258 [00:32:00:00, 8.0111/s]/c
class Images Instances P R mAP50 mAP50-95: 100% 129/129 [00:30:00:00,
all 4114 5858 0.99 0.998 0.993 0.924

200 epochs completed in 3.524 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 3.98M
Optimizer stripped from runs/train/exp/weights/best.pt, 3.98M

```

Gambar 4. Hasil Training Dataset menggunakan YOLOv5

```

epoch GPU mem box_loss cls_loss dfl_loss instances Size
199/200 2.13G 0.394 0.1014 0.855 2 640: 100% 258/258 [00:37:00:00, 6.8911/s]
Class Images Instances box/P R mAP50 mAP50-95: 100% 129/129 [00:22:00:00, 6.0211/s]

epoch GPU mem box_loss cls_loss dfl_loss instances Size
199/200 2.13G 0.393 0.1011 0.852 4 640: 100% 258/258 [00:37:00:00, 6.8111/s]
Class Images Instances box/P R mAP50 mAP50-95: 100% 129/129 [00:22:00:00, 6.0011/s]

epoch GPU mem box_loss cls_loss dfl_loss instances Size
200/200 2.13G 0.391 0.1005 0.851 2 640: 100% 258/258 [00:37:00:00, 6.8011/s]
Class Images Instances box/P R mAP50 mAP50-95: 100% 129/129 [00:22:00:00, 6.0711/s]

200 epochs completed in 3.288 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 6.39M
Optimizer stripped from runs/detect/train/weights/best.pt, 6.39M

```

Gambar 5. Hasil Training Dataset menggunakan YOLOv8

### 3.2 Perbandingan Hasil Validasi

Setelah dilakukan proses *training* dataset dengan menggunakan platform google collab, maka didapatkan hasil parameter validasi berbasis confusion matriks. Hasil lebih jelas validasi *training* dataset menggunakan YOLOv5 dan YOLOv8 disajikan pada pada Tabel 4 dan Tabel 5.

Tabel 4. Hasil *Training* Dataset YOLOv5

| YOLOv5    |              |       |          |       |
|-----------|--------------|-------|----------|-------|
| Class     | All          | Gate  | Obstacle | Drum  |
| Precision | 0.99         | 0.99  | 0.989    | 0.991 |
| Recall    | <b>0.998</b> | 0.999 | 0.998    | 0.996 |
| mAP50     | 0.993        | 0.995 | 0.993    | 0.993 |
| Map50-95  | 0.924        | 0.949 | 0.905    | 0.917 |

Tabel 5. Hasil *Training* Dataset YOLOv8

| YOLOv8    |              |       |          |       |
|-----------|--------------|-------|----------|-------|
| Class     | All          | Gate  | Obstacle | Drum  |
| Precision | <b>0.994</b> | 0.998 | 0.991    | 0.993 |
| Recall    | 0.997        | 0.997 | 0.997    | 0.996 |
| mAP50     | 0.993        | 0.995 | 0.992    | 0.993 |
| Map50-95  | <b>0.94</b>  | 0.96  | 0.928    | 0.933 |

Dilihat dari hasil Tabel 4 dan Tabel 5, pada model YOLOv8 unggul dalam parameter precision dengan nilai 0.994 dan parameter map50-95 yaitu 0.94. Parameter precision menunjukkan bahwa model YOLOv8 mampu menghasilkan pemodelan yang relatif lebih akurat jika dibandingkan dengan YOLOv5 dengan nilai 0.99 dalam memprediksi kelas positif. Sedangkan untuk parameter map50-95 digunakan sebagai indikator akurasi klasifikasi objek, ketepatan lokasi bounding box dan konsistensi model di berbagai tingkat kesulitan dengan kriteria yang lebih ketat pada variasi threshold IoU (Intersection over Unit) mulai 0.5 sampai 0.95. Dengan dasar bahwa semakin tinggi IoU, maka semakin presisi bounding box yang dituntut. Sehingga bounding box hasil prediksi model YOLOv8

lebih mendekati ground truth (label aktual) jika dibandingkan dengan YOLOv5, dengan nilai map50-95 pada YOLOv8 adalah 0.94, sedikit lebih tinggi dibandingkan YOLOv5 dengan nilai 0.924. YOLOv5 hanya unggul pada parameter recall atau kemampuan model dalam menemukan semua label positif dengan selisih yang sangat kecil yaitu 0.001 jika dibandingkan dengan YOLOv8.

## 4. Kesimpulan

Penelitian ini berhasil menganalisis perbandingan performa pelatihan YOLOv5n dan YOLOv8n pada dataset citra untuk deteksi objek pada *prototipe Autonomous Underwater Vehicle* (AUV). YOLOv8n menunjukkan keunggulan untuk parameter precision (akurasi pemodelan untuk memprediksi kelas positif) dan parameter map 50-95 (memiliki akurasi bounding box prediksi yang mendekati label aktual). Untuk waktu pelatihan menunjukkan YOLOv8n sedikit lebih lama 14 menit. Dengan arsitektur tanpa *anchor* dan modul C2f, YOLOv8n menawarkan efisiensi dan akurasi yang lebih baik, menjadikannya pilihan yang lebih optimal untuk deteksi objek real-time pada AUV, terutama untuk mendukung navigasi dan pengenalan lingkungan.

## Ucapan Terimakasih

Terima kasih kepada Politeknik Perkapalan Negeri Surabaya yang telah menyediakan fasilitas laboratorium dan dukungan teknis selama penelitian ini berlangsung.

## Daftar Rujukan

- [1] D. Laras Wati, P. Ranna, dan dan Oei Fuk Jin, "PERKEMBANGAN INTEGRASI DIGITAL TWIN DAN ROBOTIK DI INDUSTRI KONSTRUKSI," 2024.
- [2] E. Darmawan Yudi dan J. A. Yani No, "Literature Review : Implementasi Sistem Monitoring Robot Bawah Air (Underwater Robot) berbasis IoT menggunakan metode Fuzzy Logic," 2023.
- [3] I. Sobirin, J. Endrasmono, D. P. Riananda, P. Asri, M. A. Jami'in, dan I. Munadhif, "IMPLEMENTASI KONTROL PID ZIEGLER – NICHOLS UNTUK PENGENDALIAN PENYELAMAN PADA ROBOT BAWAH AIR OTONOM HYDROSHIPS," *Jurnal Elektronika dan Otomasi Industri*, vol. 11, no. 2, hlm. 409–420, Jul 2024, doi: 10.33795/elkolind.v11i2.5344.
- [4] Ii Munadhif, Widya Primaswari Putri1, M. Khoirul Hasin, Noorman Rinanto, dan Ryan Yudha Adhitya, "Kinematika Enam Derajat Kebebasan Pada Robot Bawah Air Otonom Menggunakan Pendekatan Analisa Geometri," *Jurnal Elektronika dan Otomasi Industri*, vol. 11, no. 2, hlm. 420–432, Jul 2024, doi: 10.33795/elkolind.v11i2.5341.
- [5] J. Widodo, W. Saputra, M. Allam Naufal, dan O. A. Putra, "Implementasi Yolo V8 pada Prototipe Autonomous Underwater Robot Berbasis Raspberry Pi 5 Guna Menanggulangi Pencemaran Sampah Plastik di Daerah Perairan," 2024.
- [6] J. Redmon, S. Divvala, R. Girshick, dan A. Farhadi, "Anda Hanya Melihat Sekali: Deteksi Objek Real-Time Terpadu," 2016. [Daring]. Tersedia pada: <http://pjreddie.com/yolo/>
- [7] O. Alfiano dan dan Santi Rahayu, "IMPLEMENTASI ALGORITMA DEEP LEARNING YOLO (YOU ONLY LOOK ONCE) UNTUK DETEKSI KUALITAS KENTANG SEGAR DAN BUSUK SECARA REAL TIME," *JORAPI: Journal of*

- Research and Publication Innovation*, vol. 2, no. 3, 2024, [Daring]. Tersedia pada: <https://jurnal.portalpublikasi.id/index.php/JORAPI/index>
- [8] M. A. A. Muzammil dan R. Indraswari, "Pengembangan Arsitektur Model YOLOv8 untuk Meningkatkan Performa Object Detection pada Varian Boks Warehouse Palletizing," *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, vol. 6, no. 2, hlm. 19–30, Agu 2024, doi: 10.28926/ilkomnika.v6i2.642.
- [9] Yanto, F. Aziz, dan Irmawati, "YOLO-V8 PENINGKATAN ALGORITMA UNTUK DETEKSI PEMAKAIAN MASKER WAJAH," *Jurnal Mahasiswa Teknik Informatika*, vol. 7, no. Vol. 7 No. 3 (2023): JATI Vol. 7 No. 3, hlm. 1–8, Jul 2023, doi: <https://doi.org/10.36040/jati.v7i3.7047>.
- [10] Muhammad Aqsal Sirulah Sodik, "IMPLEMENTASI ALGORITMA YOU ONLY LOOK ONCE (YOLOv5) SEBAGAI PENDETEKSI BAHASA ISYARAT INDONESIA (SIBI)," Bandung, Agu 2024. Diakses: 15 Mei 2025. [Daring]. Tersedia pada: <https://digilib.uinsgd.ac.id/94345/>
- [11] F. Romadloni dkk., "Identifikasi Warna Buoy Menggunakan Metode You Only Look Once Pada Unmanned Surface Vehicle", [Daring]. Tersedia pada: <https://journal.trunojoyo.ac.id/triac>
- [12] A. Rizal Lucky Pradana, S. Khumaidi, M. T. St, R. Andiana, dan S. St, "Identifikasi Penyebab Cacat Pada Hasil Pengelasan Dengan Image Processing Menggunakan Metode YOLO," *Jurnal Teknik Elektro dan Komputer TRIAC*, vol. 9 No.3, no. Identifikasi Penyebab Cacat Pada Hasil Pengelasan Dengan Image Processing Menggunakan Metode YOLO, hlm. 1–5, 2021, [Daring]. Tersedia pada: <https://journal.trunojoyo.ac.id/triac>
- [13] A. E. Gusti Audryadmaja, "Implementasi ROS dan Optimasi Identifikasi Warna Buoy Dengan YOLOv5 Pada Miniatur Autonomous Surface Vehicle," *Jurnal Elektronika dan Otomasi Industri*, vol. 11, no. 1, hlm. 299–308, Mei 2024, doi: 10.33795/elkolind.v11i1.5343.
- [14] M. A. Adi, Agus Khumaidi, Mohammad Basuki Rahmat, Dimas Pristovani Riananda, Muhammad Khoirul Hasin, dan Didik Sukoco, "Implementasi Sistem Deteksi Titik Api Pada area graving dock Menggunakan YOLOv5," *Jurnal Elektronika dan Otomasi Industri*, vol. 11, no. 2, hlm. 473–482, Jul 2024, doi: 10.33795/elkolind.v11i2.5233.
- [15] A. A. Wahyudi, Agus Khumaidi, Mohammad Basuki Rahmat, Dimas Pristovani Riananda, Mat Syai'in, dan Joko Endrasmono, "Implementasi Robot Operating System (ROS) Untuk Meningkatkan Akurasi Deteksi Bola Menggunakan YOLO V5 Pada KRSBI-Beroda," *Jurnal Elektronika dan Otomasi Industri*, vol. 11, no. 2, hlm. 648–661, Jul 2024, doi: 10.33795/elkolind.v11i2.5234.
- [16] Husnan, H., et al., "Deteksi objek menggunakan metode YOLO dan implementasinya pada robot bawah air," *ResearchGate*, 2023. [Online]. Available: <https://www.researchgate.net>
- [17] B. Prayoga Suswanto, M. Rahmawaty, T. Rekayasa Mekatronika, dan P. Caltex Riau, "RANCANG BANGUN SISTEM DAN MEKANISME UNDERWATER ROV (REMOTELY OPERATED VEHICLE) DEVELOPMENT OF UNDERWATER ROV (REMOTELY OPERATED VEHICLE) SYSTEMS AND MECHANISM," *Jurnal Polsri*, vol. 15, no. RANCANG BANGUN SISTEM DAN MEKANISME UNDERWATER ROV, hlm. 87–96, Okt 2023, doi: 10.53893/austenit.v15i.
- [18] Jamal, "DESAIN KONTRUKSI KAPAL RISET POLBENG BERBAHAN DASAR PLASTIK HDPE," *INOVTEK Polbeng*, vol. 14, no. 02, hlm. 130–139, Nov 2024, doi: 10.35314/be6y5813.
- [19] G. Firgiawan, N. Lintang Seina, dan P. Rosyani, "Implementasi Metode You Only Look Once (YOLO) untuk Pendeteksi Objek dengan Tools OpenCV," 2024. [Daring]. Tersedia pada: <https://jurnalmahasiswa.com/index.php/aidanspk>
- [20] Irwan Adhi Prasetya, Fadli Sukandiansyah, Novi Aryani Fitri, dan Safri Adam, "Klasifikasi kualitas buah jeruk menggunakan computer vision dengan arsitektur YOLO V8," *Jurnal Pendidikan Informatika dan Sains*, vol. 13, no. 2, hlm. 187–201, Des 2024, doi: 10.31571/saintek.v13i2.8346.
- [21] D. Triyanto, M. Zidan, M. Wahyudi, L. Pujiastuti, U. Bina Sarana Informatika, dan S. Antar Bangsa, "Pengembangan Sistem Deteksi Objek Botol Real-Time dengan YOLOv8 untuk Aplikasi Vision," *Journal Computer Science*, vol. 3, no. 1, 2024.
- [22] M. Triningsi Tamo Ama, A. Rudatyo Himamunanto, dan G. Christmass Setyawan, "Peningkatan Resolusi Citra dengan Menggunakan Metode GAN untuk Aplikasi Peningkatan Gambar," Jul 2024. doi: <https://doi.org/10.30645/kesatria.v5i3.447>.
- [23] J. Sampah, S. Implementasi di Website Bank Sampah Bersinar, dan st Rifqi Fadhila Shandi, "Penggunaan Algoritma YOLOv8 untuk Deteksi," *Telkom University*, vol. 11, no. Penggunaan Algoritma YOLOv8 untuk Deteksi Jenis Sampah: Studi Implementasi di Website Bank Sampah Bersinar, hlm. 1–7, Des 2024.
- [24] C. Xiaoming dkk., "An improved algorithm based on YOLOv5 for detecting Ambrosia trifida in UAV images," *Front Plant Sci*, vol. 15, 2024, doi: 10.3389/fpls.2024.1360419.
- [25] D. Li, T. Yang, Z. Jin, W. Si-qi, dan L. Quan-yi, "YOLOv8-EMSC: A lightweight fire recognition algorithm for large spaces," *Journal of Safety Science and Resilience*, vol. 5, no. 4, hlm. 422–431, Des 2024, doi: 10.1016/j.jnlssr.2024.06.003.