



Perancangan Sistem Pendeteksi Penyakit pada Rumput Laut Menggunakan Metode *Convolutional Neural Network*

Mhd Adi Setiawan Aritonang¹, Muhammad Abrar Masril², Deosa Chaniago³, Muhammad Marshall Al Karim⁴, Viriya Mahani Cunis⁵, Surgiwe⁶

^{1,2,3,4,5,6} Teknologi Infomasi, Teknik Komputer, Institut Teknologi Batam

¹adi@iteba.ac.id, ²abrar@iteba.ac.id, ³deosa@iteba.ac.id, ⁴marshallalkarim@gmail.com, ⁵marhancunisviriya@gmail.com, ⁶surgiwewaifi@gmail.com

Abstract

This study aims to design and build a disease detection system for seaweed. They are Healthy Seaweed, Bryozoan Crust and ice-ice using the Convolutional Neural Network (CNN) method based on the YOLO v11 architecture. Seaweed cultivation is often hampered by diseases that are detrimental to farmers. Early detection is very important. This system was developed by training the YOLO v11m model using a dataset from Roboflow consisting of training, validation, and experimental data. The training process was carried out for 100 epochs with data augmentation to increase model robustness. Model performance evaluation used precision, recall, F1-Score and mean Average Precision (mAP) metrics. The results showed good performance with an mAP50 achievement of around 0.84 on the validation data. This system was successfully implemented into an interactive web application using Flask which allows users to upload seaweed images to be visually detected for diseases.

Keywords: *Seaweed Disease Detection; Convolutional Neural Network (CNN); YOLO; Ultralytics; Flask*

Abstrak

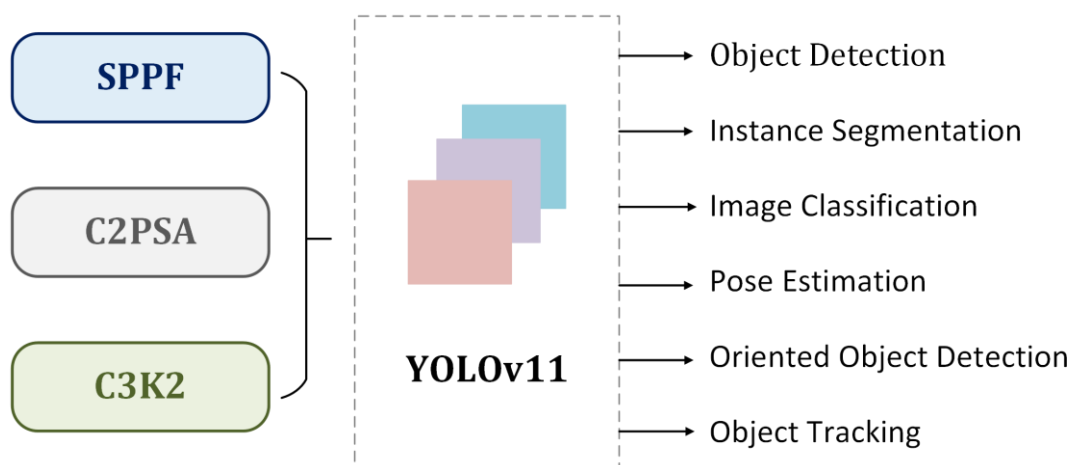
Penelitian ini bertujuan untuk merancang dan membangun sistem deteksi penyakit pada rumput laut, yaitu *Healthy Seaweed*, *Kerak Bryozoan*, dan *ice-ice*, dengan menggunakan metode *Convolutional Neural Network* (CNN) berbasis arsitektur YOLO v11. Budidaya rumput laut seringkali terkendala oleh penyakit yang merugikan petani, sehingga deteksi dini menjadi sangat penting. Sistem ini dikembangkan dengan melatih model YOLO v11m menggunakan dataset dari Roboflow yang terdiri dari data latih, validasi, dan percobaan. Proses pelatihan dilakukan sebanyak 100 *epoch* dengan augmentasi data untuk meningkatkan ketahanan model. Evaluasi performa model menggunakan metrik *precision*, *recall*, *F1-Score*, dan *mean Average Precision* (mAP). Hasil penelitian menunjukkan kinerja yang baik, dengan pencapaian mAP50 sekitar 0.84 pada data validasi. Sistem ini berhasil diimplementasikan ke dalam aplikasi web interaktif menggunakan Flask, yang memungkinkan pengguna mengunggah gambar rumput laut untuk dideteksi penyakitnya secara visual.

Kata kunci: *Deteksi Penyakit Rumput Laut; Convolutional Neural Network (CNN); YOLO v11; Ultralytics; Flask.*

© 2025 Jurnal Pustaka AI

1. Pendahuluan

Rumput laut merupakan salah satu komoditas penting dalam sektor perikanan dan kelautan di Indonesia. Namun, budidaya rumput laut seringkali terhambat oleh berbagai penyakit yang dapat menyebabkan kerugian besar bagi petani [3]. Penelitian sebelumnya telah dilakukan oleh Nurlinda Dkk [4], akan tetapi mereka menggunakan data [2] dengan metode split 80 persen data untuk pelatihan, 20 persen untuk pengujian dan menggunakan klasifikasi gambar dengan model ResNet [5]. Sedangkan di penelitian ini kami akan utilisasi deteksi objek dengan YOLO versi 11 [6], dengan melatih menggunakan data pelatihan, data validasi yang digunakan sebagai uji akurasi terhadap model yang dilatih saat berlangsung atau pasca pelatihan, dan uji coba model menggunakan data percobaan. Perbedaan dari penelitian sebelumnya, [4] menggunakan metode klasifikasi gambar, sedangkan penelitian ini merupakan deteksi obyek, yaitu dengan menunjukkan penyakit dengan ditandai oleh *bounding-box*, untuk klarifikasi letak posisi penyakit dari citra yang telah diunggah. Walaupun secara prinsip kerja sama-sama untuk mengidentifikasi fenomena citra, deteksi obyek memberikan petunjuk sehingga efektif dalam operasi pendeteksian obyek secara pandangan visual.



Gambar 1. Arsitektur YOLO 11 [6]

Arsitektur YOLO v11 [6] dirancang untuk optimalisasi kecepatan dan akurasi dengan mengembangkan inovasi dari YOLO v8 [8], v9 [10], dan v10 [11] yang berpusat pada blok C3K2, modul SPFF, dan blok C2PSA untuk pemrosesan informasi spasial yang superior dan inferensi cepat. Dibekali menggunakan blok C3K2—evolusi dari CSP *bottleneck* [9] dengan konvolusi kernel 3x3 yang lebih efisien daripada C2f pada YOLO v8 [8] untuk ekstraksi fitur di berbagai tahap, didukung oleh Blok Konvolusional dasar dan *Bottleneck* yang terinspirasi ResNet [5]. Pada bagian *neck*, YOLO v11 mempertahankan modul SPFF untuk mengumpulkan fitur multi-skala secara efektif, meningkatkan deteksi objek berukuran variatif, khususnya objek kecil. Inovasi signifikan hadir melalui mekanisme atensi dengan blok C2PSA yang mengintegrasikan *Position-Sensitive Attention (PSA)*, memungkinkan model untuk fokus pada detail penting dan wilayah relevan pada gambar. YOLO versi 11 tidak hanya untuk aplikasi deteksi obyek, tetapi juga untuk operasi segmentasi, klasifikasi gambar seperti [4], estimasi pose serta mengikut arah obyek (contoh: video, *livestream*).

2. Metode Penelitian

Pada bagian ini, terbagi menjadi empat, dimana distribusi dataset diperlukan untuk mengetahui jumlah fakta dalam dataset tersebut. Lalu, alur kerja kami disesuaikan dengan alur kerja dari dokumentasi Ultralytics [17], didalam alur kerja tersebut ada sub selanjutnya yaitu pelatihan model, dilakukan mengikuti dokumentasi [17] juga serta improvisasi metode berupa augmentasi data. Hasil model terbaik kemudian dijadikan layanan *back-end* berbasis *WebAPP*.

2.1. Distribusi Dataset

Jumlah distribusi data sebagai berikut:

Tabel 1. Distribusi data per kelas terhadap jenis Data

Jenis Data	<i>Healthy Seaweed</i>	<i>Kerak Bryozoon</i>	<i>ice-ice</i>
Pelatihan	408	678	747

Jenis Data	<i>Healthy Seaweed</i>	<i>Kerak Bryozoan</i>	<i>ice-ice</i>
Validasi	39	60	76
Percobaan	19	30	38

Dalam penelitian ini, kami menggunakan YOLO 11 dengan bobot Medium untuk pekerjaan deteksi obyek. Tiga kelas target yaitu { Kelas 0: *Healthy Seaweed*, Kelas 1: *Kerak Bryozoan*, Kelas 2: *ice-ice*}. Data pelatihan digunakan untuk melatih model dengan metode *Active Learning*, data validasi merupakan data yang digunakan untuk memastikan tingkat akurasi per tiap kelas, data percobaan adalah data yang digunakan sebagai pengujian terhadap kehandalan model. Data kami peroleh dari dataset rumput laut roboflow [2] (Diakses: 15 Mei 2025).

2.2. Alur Kerja



Gambar 2. Alur kerja penelitian

Alur pada diagram tersebut dimulai dari proses "Mulai", yang menandakan inisiasi program atau sistem. Selanjutnya, sistem akan menghitung jumlah kelas yang ada, yang kemungkinan digunakan sebagai parameter untuk proses berikutnya. Setelah jumlah kelas dihitung, langkah selanjutnya adalah melakukan proses training, yaitu tahap pelatihan model berdasarkan data yang tersedia. Setelah training selesai, sistem menghasilkan output atau "Hasil" dari proses pelatihan tersebut. Terakhir, proses diakhiri dengan langkah "Selesai", yang menandakan bahwa seluruh rangkaian proses telah tuntas dilaksanakan.

2.3. Pelatihan Model

Model yang digunakan adalah YOLO 11m dengan total 409 lapisan dan sekitar 20 juta parameter terlatih. Model ini dilatih ulang menggunakan konfigurasi:

Tabel 2. Konfigurasi training model

Jumlah Epoch	100
Ukuran Gambar	640 640
Ukuran Batch	16
Perangkat Komputasi	Google Colab PRO, NVIDIA Tesla T4
Augmentasi Data	Rotasi acak maksimum $\pm 10^\circ$, Tanpa <i>flipping vertical</i>

Proses pelatihan menggunakan fungsi kehilangan deteksi dari Ultralytics dengan optimisasi otomatis menggunakan algoritma *AdamW* [18]. Parameter inisialisasi *learning rate* (0.001429), *momentum* (0.9), dan *weight decay* secara otomatis ditentukan oleh pustaka Ultralytics berdasarkan validasi awal. Pelatihan menggunakan *Automatic Mixed Precision (AMP)*.

2.4. Evaluasi

Untuk mengevaluasi performa model YOLOv11m dalam mendeteksi dan mengklasifikasikan penyakit pada rumput laut, digunakan sejumlah metrik evaluasi berbasis deteksi objek, yakni *Precision*, *Recall*, *F1-Score*, dan *mean Average Precision* (mAP).

Berikut rumus-rumus utama yang digunakan dalam evaluasi performa model:

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

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

$$F1 = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (3)$$

$$mAP@0.5 = \frac{1}{n} \sum_{i=1}^n AP_i \quad (4)$$

Dimana:

TP = True Positive

TN = True Negative

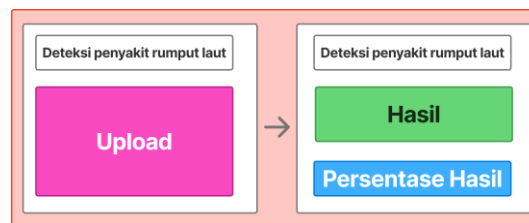
FP = False Positive

FN = False Negative

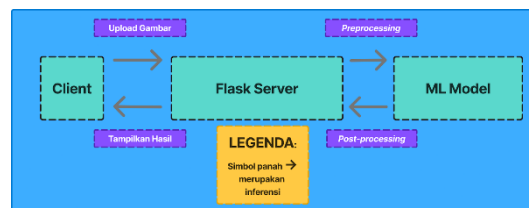
AP = Average Precision

Hasil rumus tersebut digunakan untuk memvisualisasi matriks konfusi per tiap kelas terhadap data validasi dan pengujian.

2.5. WebAPP



Gambar 3. Diagram alir UI aplikasi WebAPP

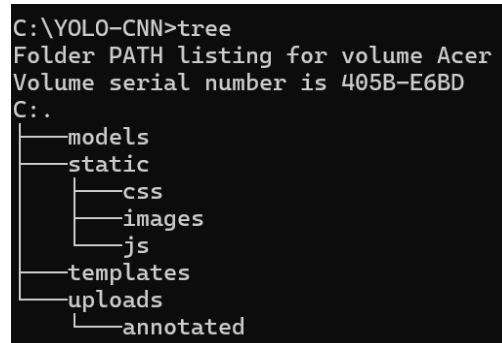


Gambar 4. Rencana Deployment dan Hosting Local menggunakan FlaskAPP

Kami menggunakan library Ultralytics YOLO [17], NumPy [12], Python [13] dan Flask [14], uuid, pillow [15], OpenCV [16] dan OS/shutil untuk deployment Aplikasi Web. *Front-end*, kami menggunakan HTML, CSS dan JS. Frontend menyediakan form HTML dengan input tipe file yang memungkinkan pengguna memilih gambar dari perangkat lokal mereka. *Backend* Flask memiliki *endpoint* yang menerima *request* POST berisi data file gambar. Dilakukan validasi ekstensi file yang diizinkan (.png, .jpg, .jpeg, .webp). Gambar yang valid disimpan sementara di server (dalam *folder* uploads/) dengan nama *file* unik yang dihasilkan oleh uuid untuk menghindari konflik.

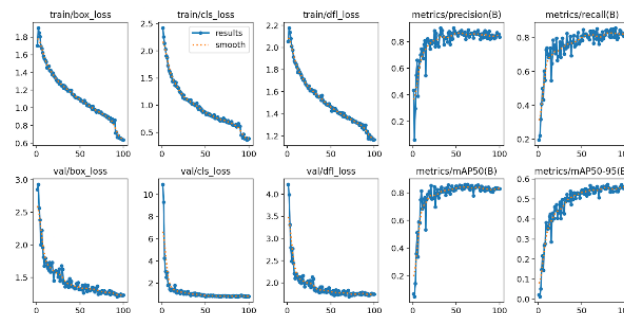
Untuk Proses deteksi *Backend*, Gambar yang telah disimpan kemudian diproses oleh model YOLOv11, dengan menggunakan library Ultralytics secara langsung, *Preprocessing input* gambar dan *postprocessing output*

(termasuk denormalisasi koordinat untuk daftar teks) ditangani sebagian besar oleh library. Gambar hasil dengan anotasi visual dapat dihasilkan menggunakan metode `.plot()` dari objek hasil prediksi Ultralytics, yang kemudian disimpan sebagai *file* gambar baru. Kami menggunakan *checkpoint* dari file (.pt) yang dihasilkan dari pelatihan.



Gambar 5. Struktur folder Model

3. Hasil dan Pembahasan



Gambar 5. Hasil pelatihan dari library Ultralytics

Gambar diatas merupakan hasil akhir *training* dari *library* ultralytics, dan dari model tersebut, kami menguji hasil yang bernama “best.pt” di dalam *folder* yang kami unduh dari hasil generasi model dari pustaka ultralytics untuk di *deploy* ke dalam WebAPP flask.

Temuan yang kami tampilkan berupa hasil pemrosesan uji evaluasi metrik data validasi serta data percobaan, menampilkan konfusi matriks yang digunakan sebagai indikator keakuratan model yang telah dilatih. Adapun WebAPP yang telah dibangun untuk merepresentasikan visual *bounding box* hasil deteksi obyek.

3.1. Uji Evaluasi Metrik Data Validasi

Setelah pemrosesan selesai, kami *mendownload file* dari hasil pemrosesan *library* ultralytics.



Gambar 6. Hasil konfusi matriks pelatihan terhadap data Validasi (atas) serta normalisasi data (bawah)

Dari temuan kami menunjukkan adanya data *background* dari hasil matriks konfusi, representasi dari segi total kelas maupun normalisasi data, secara visual, keseluruhan menunjukkan diagonal, mengindikasikan keakuratan tinggi untuk validasi data dari hasil prediksi model.

Perhitungan data validasi pasca pelatihan *Healthy Seaweed*:

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

$$= \frac{40}{(40 + 13)}$$

$$= \frac{40}{53} \approx 0.75$$

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

$$= \frac{40}{(40 + 3)}$$

$$= \frac{40}{43} \approx 0.93$$

$$\begin{aligned} F1 - Score &= 2 \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \\ &= 2 \frac{(0.7547 \times 0.9302)}{(0.7547 + 0.9302)} \approx 0.83 \\ &= \frac{40}{43} \approx 0.93 \end{aligned}$$

Perhitungan data validasi pasca pelatihan *Kerak Bryozoan*:

$$\begin{aligned} \text{Precision} &= \frac{58}{(58 + 10)} \\ &= \frac{58}{68} \approx 0.85 \\ \text{Recall} &= \frac{58}{(58 + 5)} \\ &= \frac{58}{63} \approx 0.92 \\ F1 - Score &= 2 \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \\ &= 2 \frac{(0.8529 \times 0.9206)}{(0.8529 + 0.9206)} \approx 0.88 \end{aligned}$$

Perhitungan data validasi pasca pelatihan *Ice-ice*:

$$\begin{aligned} \text{Precision} &= \frac{72}{(72 + 6)} \\ &= \frac{72}{78} \approx 0.92 \\ \text{Recall} &= \frac{72}{(72 + 21)} \\ &= \frac{72}{93} \approx 0.77 \\ F1 - Score &= 2 \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \\ &= 2 \frac{(0.9231 \times 0.7742)}{(0.9231 + 0.7742)} \approx 0.842 \end{aligned}$$

Mean Average Precision:

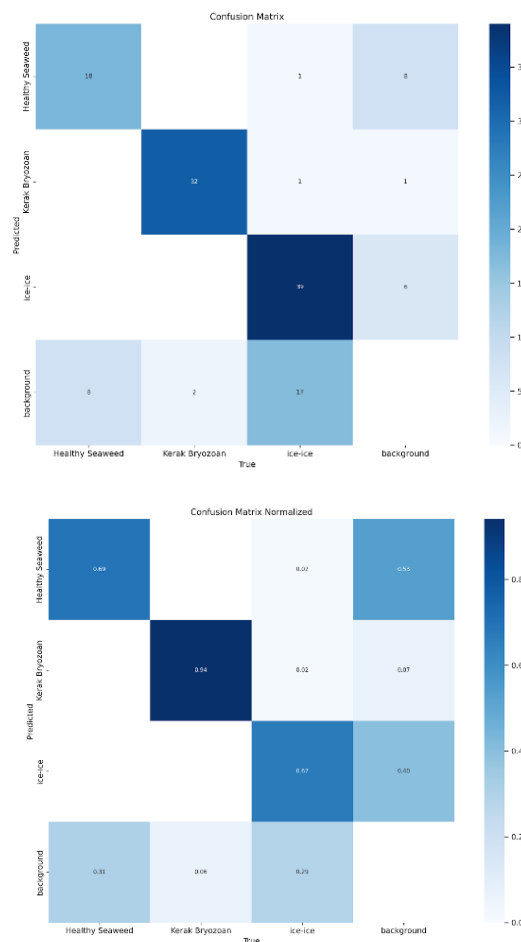
$$\text{mAP50} = \frac{(\text{Precision Healthy Seaweed} + \text{Precision Kerak Bryozoan} + \text{Precision Ice - ice})}{3}$$

$$= \frac{(0.7547 + 0.8529 + 0.9231)}{3}$$

$$= \frac{2.5307}{3} \approx 0.84$$

Berdasarkan rerataan presisi, kami mendapatkan di nilai 84% dari segi performa, Kelas *Healthy Seaweed* mendapatkan skor tertinggi untuk *F1-Score*, disusul oleh *Kreak Bryozoan* dan yang paling rendah ialah *Ice-Ice*. Dari segi *Recall*, dimana model menentukan obyek secara relevan, *Healthy Seaweed* mendapatkan skor tertinggi dan *Ice-Ice* mendapatkan skor terendah. Untuk nilai presisi, *Ice-Ice* mendapatkan nilai tertinggi dan *Healthy Seaweed* mendapatkan nilai terendah.

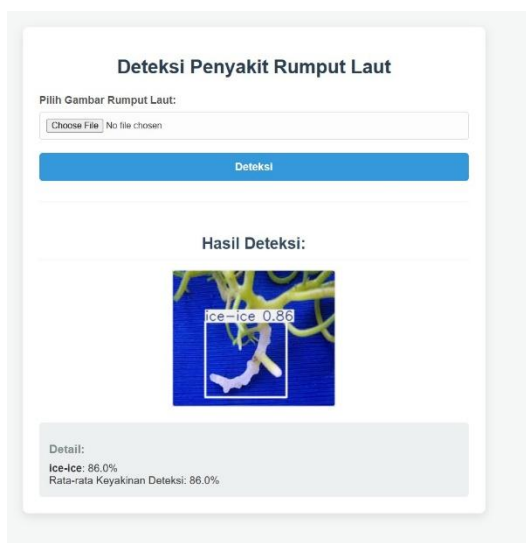
3.2. Uji Evaluasi Metrik data Percobaan



Gambar 7. Hasil konfusi matriks pelatihan terhadap data percobaan (atas) serta normalisasi data (bawah)

Terdapat keanehan dalam hasil komputasi matriks konfusi untuk data percobaan, dikarenakan data *kerak bryozoan* memiliki kelebihan dua data (dalam 30 jumlah data), sedangkan *ice-ice* kelebihan satu data (dalam 38 jumlah data). Walaupun hasil *recall* dari konfusi matriks terlihat sempurna, ini menjadi anomali kami temukan, kami berspekulasi terhadap *library* ultralytics, atau penggenerasian data otomatis. Kendati demikian, hasil percobaan sempurna untuk kelas *healthy Seaweed* dikarenakan sempurna untuk pendeteksi kelas objek tersebut. Tetapi masih perlu diuji lagi untuk kelas *ice-ice*.

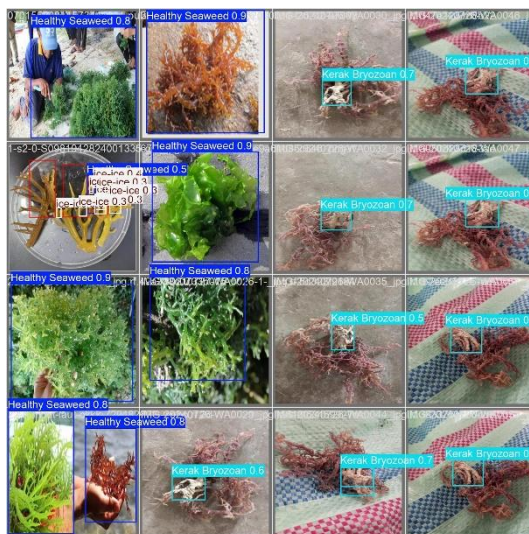
3.3. Deployment WepAPP Flask



Gambar 8. Hasil pembangunan WebAPP menggunakan flask

Kami mendapatkan foto penyakit rumput laut “ice-ice” tersebut di *website the conversation* [1] (Diakses: 31 mei 2025). Dan model kami mendeteksi dengan benar dan memberikan detail tingkat kepercayaan (*confidence*) sebesar 86%.

4. Kesimpulan



Gambar 9. Hasil deteksi dari sampel prediksi label, di generasikan oleh *library ultralytics*

Penelitian ini berhasil mengembangkan sebuah sistem untuk mendeteksi penyakit pada rumput laut, yaitu “Healthy Seaweed”, “Kerak Bryozoan”, dan “ice-ice”, dengan memanfaatkan metode *Convolutional Neural Network* (CNN) [19] berbasis arsitektur YOLO 11m. Model yang dilatih menggunakan dataset dari Roboflow menunjukkan kinerja yang baik, dengan pencapaian metrik *mean Average Precision* (mAP 50) sekitar 0.84 pada data validasi, serta nilai presisi, *recall*, dan *F1-Score* yang cukup tinggi untuk masing-masing kelas penyakit. Walaupun pada pengujian data percobaan ditemukan beberapa anomali pada hasil matriks konfusi untuk kelas “Kerak Bryozoan” dan “ice-ice” yang mengindikasikan perlunya investigasi dan pengujian lebih lanjut, deteksi untuk kelas “Healthy Seaweed” menunjukkan hasil yang sempurna. Lebih lanjut, sistem ini telah berhasil diimplementasikan menjadi aplikasi web interaktif menggunakan Flask, yang mampu melakukan deteksi penyakit secara visual dari gambar yang diunggah pengguna. Secara keseluruhan, penelitian ini menunjukkan bahwa YOLO 11m memiliki potensi signifikan sebagai alat bantu efektif untuk identifikasi penyakit rumput laut, namun pengujian lebih mendalam disarankan untuk meningkatkan keandalan deteksi pada semua variasi kondisi dan kelas penyakit. Dibuktikan di bagian 3.3 dimana sampel foto acak dari internet (*Ice-Ice*) yang kemudian di uji untuk melihat *output* model. Dari hasil pemrosesan tersebut, dapat membantu pengguna untuk mengetahui Lokasi pasti penyakit yang di idap oleh rumput laut berdasarkan prediksi model.

Daftar Rujukan

- [1] <https://theconversation.com/budi-daya-perikanan-indonesia-tumbuh-pesat-ini-3-caranya-agar-tak-merusak-lingkungan-201674> (Diakses 2 Juni 2025)
- [2] Rifki Andre, Dataset Rumput Laut, Roboflow (2024). Available at <https://universe.roboflow.com/rifki-andre-2024/penyakit-rumput-laut>
- [3] Munaeni, W., Lesmana, D., Irawan, H., Hamka, M.S. and Nafsiyah, I., 2023. Potensi Budidaya dan Olahan Rumput Laut di Indonesia. Tohar Media.
- [4] Nurlinda, N., Hasmin, E. and Jufri, J., 2024. DETEKSI PENYAKIT RUMPUT LAUT DENGAN RESIDUAL NEURAL NETWORK. Jurnal Tekinkom (Teknik Informasi dan Komputer), 7(2), pp.637-646.
- [5] Wu, W., Huo, L., Yang, G., Liu, X., & Li, H. (2025). Research into the application of ResNet in soil: A review. Agriculture, 15(6), 661.
- [6] Khanam, R. and Hussain, M. (2024) YOLOv11: An Overview of the Key Architectural Enhancements. arXiv:2410.17725 [cs.CV]. Available at: <https://arxiv.org/abs/2410.17725>
- [7] Visualiasi YOLO model V11, (2024). Available at <https://medium.com/@nikhil-rao-20/yolov11-explained-next-level-object-detection-with-enhanced-speed-and-accuracy-2dbe2d376f71> (Diakses 18 Juni 2025)
- [8] Ultralytics (2023) YOLOv8. GitHub. Available at: <https://github.com/ultralytics/ultralytics> (Accessed: 2 June 2025).
- [9] Yan, C., & Li, H. (2025). CSPNet: A feature interaction network for tomato leaf disease detection in complex scenarios. Computers and Electronics in Agriculture, 238, 110823.
- [10] Wang, C.-Y., Liao, H.-Y. M., Yeh, I.-H., Wu, S.-H., Chen, P.-Y. and Hsieh, J.-W. (2024) YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. arXiv:2402.13616 [cs.CV]. Available at: <https://arxiv.org/abs/2402.13616> (Accessed: 2 June 2025).
- [11] Wang, A., Wang, L., Li, J., Song, L., Liu, Y., Wang, L. and Zhang, Z. (2024) YOLOv10: Real-Time End-to-End Object Detection. arXiv:2405.14458 [cs.CV]. Available at: <https://arxiv.org/abs/2405.14458> (Accessed: 2 June 2025).
- [12] NumPy committers, "NumPy," NumPy, 2025. [Online]. Available: <https://numpy.org>. (Accessed: 18 June 2025).
- [13] Python Software Foundation (2025) Python. Available at: <https://www.python.org> (Accessed: 18 June 2025).
- [14] Pallets Projects (2025) Flask. Available at: <https://flask.palletsprojects.com> (Accessed: 18 June 2025).
- [15] Pillow-Contributors (2025) Pillow. Available at: <https://pypi.org/project/Pillow/> (Accessed: 18 June 2025).
- [16] OpenCV team (2025) OpenCV (Open Source Computer Vision Library). Available at: <https://opencv.org> (Accessed: 18 July 2025).
- [17] Ultralytics (2025) Ultralytics YOLO. Available at: <https://github.com/ultralytics/ultralytics> (Accessed: 18 June 2025).
- [18] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," arXiv preprint arXiv:1711.05101, 2019. [Online]. Available: <https://arxiv.org/abs/1711.05101>. (Accessed: July 18, 2025).
- [19] Adilova, L., & Geiger, B. (2025). Geometric and Information-Theoretic Compression in Neural Classifier Training. In Book of Abstracts Workshop on AI for Sustainable Energy Systems and Green AI (pp. 49-51).
- [20] Putra, A. D., & Saputra, G. E. (2025). Implementation of YOLOv8 Algorithm for Web-Based Detection of Coffee Fruit Ripeness. Journal of Artificial Intelligence and Software Engineering, 5(2), 432-445.
- [21] Sidhu, M., Chopra, H., Blume, A., Kim, J., Reddy, R. G., & Ji, H. (2025). Search and detect: Training-free long tail object detection via web-image retrieval. In Proceedings of the Computer Vision and Pattern Recognition Conference (pp. 15129-15138).