



## Deteksi Ikan Menggunakan Metode Faster R-CNN

Yolanda Eka Pratiwi<sup>1</sup>, Yurni Oktarina<sup>2</sup>, Tresna Dewi<sup>3</sup>

<sup>1,2,3</sup>Teknik Elektro, Politeknik Negeri Sriwijaya

<sup>1</sup>pratiwiyolandaeka@gmail.com, <sup>2</sup>yurnioktarina@polsri.ac.id\*, <sup>3</sup>tresna\_dewi@polsri.ac.id

### Abstract

Automatic fish detection in video is a challenging task in the field of computer vision, which can be addressed using deep learning methods. This study proposes the use of the Faster Region-based Convolutional Neural Network (Faster R-CNN) to detect two types of fish, namely Manfish and Lemonfish, in video data. The dataset was constructed by extracting frames from video and processing them using the Roboflow platform. The model was trained and tested using pre-split training and testing sets. The training process was conducted over 40 epochs using the Adam optimization algorithm to improve detection accuracy. Model evaluation was carried out using several metrics, including Precision, Recall, F1-score, mean Average Precision (mAP), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared ( $R^2$ ). The results show that the model achieved a precision of 94% for the Lemonfish class, and a precision of 95% for the Manfish class with a *mean Average Precision* (mAP) of 57%. Although the model demonstrates good performance, there is still room to improve accuracy through further optimization.

Keywords: Fish Detection, Deep Learning, Faster R-CNN, Computer Vision, Model Evaluation.

### Abstrak

Deteksi ikan dalam video secara otomatis merupakan tantangan dalam bidang *computer vision* yang dapat diselesaikan menggunakan metode *deep learning*. Penelitian ini mengusulkan penggunaan *Faster Region-based Convolutional Neural Network* (Faster R-CNN) untuk mendeteksi dua jenis ikan, yaitu Manfish dan Lemonfish, dalam video. Dataset diperoleh dengan mengekstraksi frame dari video dan memprosesnya menggunakan platform Roboflow. Model dilatih dan diuji dengan menggunakan *training* dan *testing set* yang telah disiapkan. Proses pelatihan dilakukan selama 40 epoch dengan algoritma optimasi Adam untuk meningkatkan akurasi deteksi. Evaluasi model dilakukan menggunakan berbagai metrik, seperti *Precision*, *Recall*, *F1-score*, *mean Average Precision* (mAP), *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), dan *R-squared* ( $R^2$ ). Hasil penelitian menunjukkan bahwa model mencapai presisi sebesar 94% untuk kelas Lemonfish dan presisi sebesar 95% untuk kelas Manfish, dengan mAP sebesar 57%. Meskipun model menunjukkan kinerja yang baik, masih terdapat peluang untuk meningkatkan akurasi dengan optimalisasi lebih lanjut.

Kata kunci: Deteksi Ikan, *Deep Learning*, Faster R-CNN, *Computer Vision*, Evaluasi Model.

Diterima Redaksi : 23-11-2024 | Selesai Revisi : 25-12-2024 | Diterbitkan Online : 31-12-2024

### 1. Pendahuluan

Teknologi *Deep Learning* telah berkembang pesat dalam beberapa tahun terakhir dan memiliki berbagai aplikasi dalam analisis citra dan video. Salah satu penerapannya adalah dalam bidang akuakultur dan biologi kelautan, di mana deteksi ikan secara otomatis dalam video menjadi kebutuhan penting untuk keperluan monitoring, klasifikasi, dan manajemen populasi ikan di lingkungan perairan maupun akuarium [1].

Salah satu metode yang banyak digunakan dalam pendeteksian objek adalah *Faster R-CNN*, yang dikembangkan untuk meningkatkan akurasi dan efisiensi dalam pengenalan objek dibandingkan dengan

metode sebelumnya seperti R-CNN dan Fast R-CNN. *Faster R-CNN* dikembangkan oleh Ren et al. (2015) sebagai peningkatan dari metode pendahulunya, yaitu R-CNN dan Fast R-CNN, dengan mengintegrasikan proses *region proposal* langsung ke dalam jaringan convolutional, sehingga meningkatkan efisiensi dan akurasi secara signifikan [2].

Berbagai studi sebelumnya telah menunjukkan bahwa *Faster R-CNN* mampu memberikan hasil yang unggul dalam pendeteksian objek, terutama dalam konteks kompleks seperti pendeteksian hewan laut [3]. Misalnya, penelitian oleh Islam et al. (2020) menggunakan *Faster R-CNN* untuk mendeteksi spesies ikan di laut dan menunjukkan akurasi yang tinggi

dalam kondisi pencahayaan dan posisi ikan yang bervariasi [4].

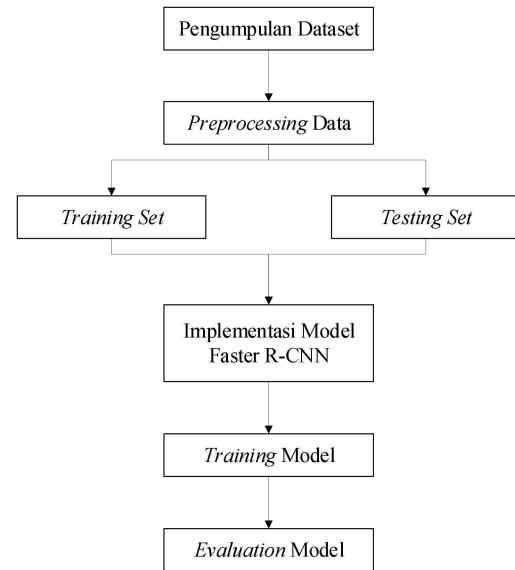
Meskipun demikian, tantangan seperti kebutuhan akan dataset yang besar dan beragam, serta kecepatan pemrosesan yang tinggi, tetap menjadi fokus penelitian lanjutan. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem deteksi ikan dalam video dengan memanfaatkan Faster R-CNN untuk mendeteksi dua jenis ikan, yaitu Manfish dan Lemonfish. Melalui pendekatan ini, diharapkan sistem yang dikembangkan dapat secara otomatis mengidentifikasi ikan dalam video dengan tingkat akurasi yang tinggi serta potensi penerapan pada skala industri maupun penelitian ilmiah kelautan.

## 2. Metode Penelitian

Penelitian ini menggunakan model *Faster Region-based Convolutional Neural Network* (Faster R-CNN) dengan backbone MobileNetV3 untuk mendeteksi ikan pada video. Model Faster R-CNN dipilih karena kemampuannya dalam mendeteksi objek secara akurat dengan memanfaatkan *Convolutional Neural Network* (CNN) untuk mengekstraksi fitur-fitur penting [2]. Sementara itu, backbone MobileNetV3 dipilih karena memiliki tingkat akurasi yang lebih tinggi dibandingkan dengan MobileNetV2. Selain itu, MobileNetV3 juga memiliki jumlah parameter dan operasi komputasi yang jauh lebih sedikit dibandingkan backbone besar seperti ResNet atau VGG, tanpa mengorbankan performa deteksi secara signifikan, sehingga menjadikannya pilihan ideal dalam implementasi sistem deteksi objek yang ringan namun tetap akurat [5].

Penelitian ini diawali dengan pengumpulan dataset berupa video yang menampilkan dua jenis ikan, yaitu Manfish dan Lemonfish. Video yang diperoleh kemudian melalui tahap *preprocessing*, di mana setiap frame diekstrak dan dikonversi menjadi gambar menggunakan platform Roboflow. Setelah proses konversi selesai, dataset dibagi menjadi dua bagian yaitu *training set* untuk melatih model dan *testing set* untuk menguji performanya.

Dataset yang telah diproses selanjutnya digunakan untuk melatih dan menguji model Faster R-CNN. Pada tahap akhir, dilakukan evaluasi kinerja model untuk menganalisis tingkat akurasi serta efektivitas deteksi objek. Hasil evaluasi ini menjadi dasar dalam menilai keberhasilan model yang telah dikembangkan. Berikut alur penelitian yang ditunjukkan pada Gambar 1.



Gambar 1. Alur Penelitian

### 2.1. Dataset

Penelitian ini menggunakan dataset berupa video yang menampilkan ikan Manfish dan Lemonfish dalam satu frame di dalam akuarium. Video tersebut kemudian diproses menggunakan platform Roboflow untuk diekstrak menjadi gambar. Dari hasil proses ini, diperoleh 300 gambar yang digunakan sebagai dataset dalam penelitian. Setiap gambar telah diberi anotasi berupa bounding box untuk menandai posisi ikan dalam gambar secara manual, menggunakan fitur labeling yang tersedia pada Roboflow.

Penggunaan dataset berbasis video untuk ekstraksi frame dalam penelitian ini memiliki beberapa keunggulan. Dibandingkan dengan pengambilan gambar statis satu per satu, metode ini memungkinkan penangkapan berbagai posisi, orientasi, serta interaksi ikan secara alami, sehingga memperkaya variasi data dan meningkatkan kemampuan generalisasi model. Hal ini sejalan dengan penelitian oleh Siddiqui et al. (2018), yang menunjukkan bahwa data visual yang diperoleh dari cuplikan video dapat menghasilkan model deteksi objek yang lebih kuat dan tahan terhadap variasi lingkungan [6].

Sampel yang diambil dari dataset penelitian ini ditunjukkan pada Gambar 2.



Gambar 2. Dataset

## 2.2. Preprocessing Data

Video ikan yang diambil terlebih dahulu dikonversi menjadi gambar menggunakan Roboflow. Sebelum proses konversi, video disesuaikan dengan pengaturan 1 frame per detik, menghasilkan total 300 gambar sebagai dataset.

Setiap gambar kemudian diberi anotasi atau label sesuai dengan objek yang ada di dalamnya, yaitu "Manfish-6cm" dan "Lemonfish-3cm", di mana 6 cm dan 3 cm menunjukkan panjang masing-masing ikan. Anotasi ini berfungsi untuk menandai posisi objek dalam gambar, sehingga memudahkan model dalam mengenali dan membedakan kedua jenis ikan.

Setelah proses anotasi selesai, gambar-gambar tersebut diubah ukurannya (*resize*) menjadi 640x640 piksel untuk menyeragamkan format input data. Selanjutnya, dataset dibagi menjadi dua bagian utama, yaitu *training set* untuk melatih model dan *test set* untuk menguji performanya. Dataset dibagi menjadi *training set* (80%) dan *testing set* (20%). Proporsi pembagian dataset mengacu pada praktik umum dalam deep learning, yaitu sekitar 80:20 atau 70:30 [7].

## 2.3. Faster Region-based Convolutional Neural Network

*Faster Region-based Convolutional Neural Network* (Faster R-CNN) adalah salah satu metode *object detection* yang populer dan efisien. Metode ini merupakan pengembangan dari R-CNN dan Fast R-CNN dengan peningkatan signifikan dalam kecepatan dan akurasi. Menurut Ren et al. (2017), Faster R-CNN bekerja dengan tiga tahap utama. Pertama, gambar dimasukkan ke dalam *Convolutional Neural Network*

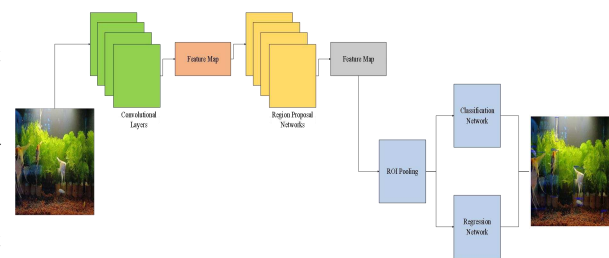
(CNN) untuk mengekstraksi fitur, menghasilkan *feature map* yang digunakan untuk mendeteksi objek.

Selanjutnya, tahap *Region Proposal Network* (RPN) digunakan untuk menghasilkan wilayah kandidat (*region proposals*) yang berpotensi berisi objek. Berbeda dari metode sebelumnya yang menggunakan *selective search*, RPN secara langsung menghasilkan proposal objek dengan lebih cepat dan efisien. RPN menggunakan *anchor boxes* untuk mendeteksi objek dengan berbagai ukuran dan rasio aspek.

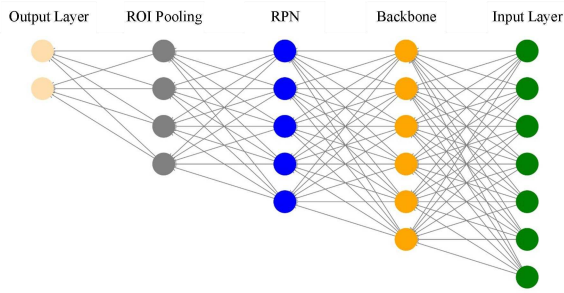
Setelah wilayah kandidat diperoleh, tahap selanjutnya adalah *ROI Pooling & Classification*. Di sini, setiap *region proposal* diproses untuk menentukan kelas objek yang terdeteksi dan memperbaiki posisi *bounding box* agar lebih akurat [2]. Menurut Liu et al. (2020) dalam publikasi *A Survey of Object Detection Based on CNNs*, modul ROI Pooling memungkinkan fitur yang berbeda ukuran dikonversi menjadi ukuran tetap, memungkinkan klasifikasi dan regresi posisi dilakukan secara efisien oleh fully connected layers di tahap akhir [8].

Menurut Huang et al. (2017), Faster R-CNN menawarkan keseimbangan yang baik antara kecepatan dan akurasi, terutama ketika digunakan bersama backbone jaringan yang kuat seperti ResNet atau Inception [9]. Penelitian lain oleh Zhang et al. (2020) menunjukkan bahwa integrasi Faster R-CNN dengan teknik augmentasi data dan pelatihan end-to-end mampu meningkatkan kinerja deteksi secara signifikan pada berbagai dataset benchmark [10].

Faster R-CNN juga telah berhasil diterapkan dalam berbagai bidang, termasuk pengenalan wajah [11] serta deteksi objek medis [12].



Gambar 3. Arsitektur Faster R-CNN



Gambar 4. Hidden Layers of Faster R-CNN

#### 2.4. Model Training

*Model Training* adalah proses penting dalam *deep learning*, di mana sebuah model, seperti Faster R-CNN, dilatih menggunakan dataset beranotasi agar mampu mengenali pola dalam data dan melakukan deteksi objek secara akurat. Pada tahap ini, model menerima data berupa gambar yang telah diberi label (*bounding box* dan kelas objek) lalu mempelajari fitur-fitur yang membedakan antara satu kelas objek dengan yang lainnya.

Dalam proyek ini, Faster R-CNN digunakan karena merupakan salah satu algoritma *object detection* yang terbukti efektif dalam berbagai studi. Faster R-CNN mengintegrasikan *Region Proposal Network* (RPN) dengan jaringan klasifikasi dan regresi *bounding box* dalam satu arsitektur end-to-end [2].

Selama proses pelatihan, model menggunakan algoritma optimasi Adam [13], yang dikenal karena kemampuannya dalam menangani data berskala besar dan parameter model yang tinggi. Adam menggabungkan kelebihan dari *Adaptive Gradient Algorithm* (AdaGrad) dan *Root Mean Square Propagation* (RMSProp), yang memungkinkan pembaruan bobot yang lebih efisien selama proses training.

Proses pelatihan dilakukan selama 40 epoch dan menggunakan batch size 4. Batch size 4 dipilih karena model Faster R-CNN tergolong kompleks dan membutuhkan banyak memori GPU, terutama saat memproses data resolusi tinggi. Dengan batch size yang kecil, pelatihan tetap dapat dilakukan tanpa kehabisan memori, sekaligus memberikan variasi gradien yang lebih besar sehingga membantu model dalam menghindari jebakan local minima dan meningkatkan kemampuan generalisasi, terutama ketika dataset yang digunakan relatif kecil. Sementara itu, jumlah epoch sebanyak 40 bertujuan untuk memastikan bahwa model memiliki cukup kesempatan untuk belajar dari seluruh data secara menyeluruh. Pada dataset yang hanya terdiri dari 300 gambar, model perlu melihat data berulang kali agar mampu memahami pola dan karakteristik objek yang ingin dikenali. Dengan 40 epoch dan batch size 4, model akan menjalani sekitar 3.000 iterasi selama pelatihan, yang secara empiris

cukup untuk mencapai konvergensi pada banyak kasus deteksi objek dengan skala dataset menengah. Oleh karena itu, konfigurasi ini dianggap seimbang antara keterbatasan perangkat keras dan kebutuhan pelatihan yang optimal.

Pada setiap epoch, model mengevaluasi kinerjanya dengan menghitung *training loss* dan *validation loss*. *Loss function* yang umum digunakan dalam Faster R-CNN mencakup gabungan antara *classification loss* (misalnya *cross-entropy*) dan *regression loss* (misalnya *smooth L1 loss*) [14].

Selama proses training, Google Colab akan merekam seluruh aktivitas yang terjadi dan menyimpan hasil pelatihan ke Google Drive. File hasil pelatihan disimpan dalam format **best.pt**, yang nantinya digunakan untuk menguji model dengan dataset pengujian (*testing dataset*).

Setelah model selesai dilatih, model tersebut dapat digunakan untuk menguji keakuratannya dalam mendeteksi ikan Manfish dan Lemonfish. Pengujian dilakukan menggunakan video yang berisi kedua jenis ikan tersebut. Tujuan utama dari training model adalah menghasilkan model yang dapat mendeteksi objek dengan akurasi tinggi dan mampu menggeneralisasi pola dari data yang diberikan.

#### 2.5. Model Evaluation

Model evaluasi adalah proses mengukur kinerja model yang telah dilatih menggunakan dataset. Dalam penelitian ini, evaluasi model dilakukan dengan menggunakan beberapa parameter, yaitu *Confusion Matrix* [15], *Precision*, *Recall*, *F1-score*, dan *mAP* [16] serta *Mean Squared Error* (MSE) [17], *Root Mean Squared Error* (RMSE) [18], dan *R-squared* ( $R^2$ ) [19].

Evaluasi model bertujuan untuk menilai seberapa baik model dalam menjalankan tugas yang diberikan. Selain itu, evaluasi juga berperan dalam menentukan apakah model memerlukan penyempurnaan lebih lanjut agar menghasilkan performa yang lebih optimal.

Menurut Gosiewska et al. (2022), evaluasi yang komprehensif menggunakan berbagai metrik sangat penting untuk memastikan bahwa model tidak hanya overfit pada data pelatihan, tetapi juga mampu menggeneralisasi dengan baik pada data baru [20]. Penelitian lain oleh Miller et al. (2024) menekankan pentingnya memahami kelebihan dan kekurangan dari setiap metrik evaluasi, serta potensi bias yang dapat mempengaruhi interpretasi hasil [21]. Powers (2020) juga mengkritisi penggunaan metrik tradisional seperti *precision* dan *recall* tanpa mempertimbangkan bias dan menyarankan penggunaan metrik seperti *informedness* dan *markedness* untuk evaluasi yang lebih akurat [22].

### 3. Hasil dan Pembahasan

Setelah proses training model Faster R-CNN selesai, model dapat digunakan untuk mendeteksi objek dalam video, khususnya untuk mengenali jenis ikan. Saat video diproses, model akan menampilkan *bounding box* di sekitar objek yang terdeteksi serta menampilkan nama jenis ikan yang sesuai.

Hasil pendeteksian ini dapat dilihat pada gambar berikut, di mana model telah berhasil mengidentifikasi ikan dalam video dengan menandai setiap objek yang terdeteksi menggunakan *bounding box* dan label nama yang sesuai.



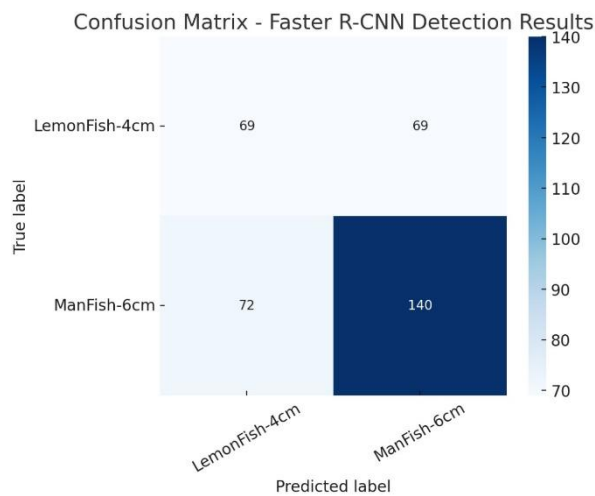
Gambar 5. Hasil Pendeteksian Video Menggunakan Model Faster R-CNN

#### 3.1. Confusion Matrix

Tabel 1. Hasil *Confusion Matrix*

	LemonFish-4cm	ManFish-6cm
LemonFish-4cm	69	69
ManFish-6cm	72	140

Tabel 1 menampilkan hasil *confusion matrix* pada saat pengujian dataset. Sedangkan gambar dibawah merupakan visualisasi *confusion matrix* untuk memberikan gambaran yang lebih jelas mengenai performa model.



Gambar 6. *Confusion Matrix*

Gambar 6 menunjukkan hasil evaluasi kinerja model deteksi terhadap dua kelas yang berbeda, yaitu "LemonFish-4cm" dan "ManFish-6cm". Berikut adalah hasil yang diperoleh:

- Sebanyak 69 data dari kelas "LemonFish-4cm" berhasil diprediksi dengan benar oleh model sebagai "LemonFish-4cm" (*True Positive*).
- Sebanyak 69 data dari kelas "LemonFish-4cm" salah diprediksi oleh model sebagai "ManFish-6cm" (*False Negative*).
- Sebanyak 72 data dari kelas "ManFish-6cm" salah diprediksi oleh model sebagai "LemonFish-4cm" (*False Negative*).
- Sebanyak 140 data dari kelas "ManFish-6cm" berhasil diprediksi dengan benar oleh model sebagai "ManFish-6cm" (*True Positive*).

Tabel 2. Hasil Deteksi

	Precision	Recall	F1-score	AP	mAP
LemonFish-4cm	94%	92%	91%	87%	57%
ManFish-6cm	95%	94%	94%	89%	

Berdasarkan Tabel 2, dapat disimpulkan bahwa hasil deteksi ikan dengan dua kelas, yaitu "LemonFish-4cm" dan "ManFish-6cm", menggunakan model Faster R-CNN dengan backbone MobileNet V3 menunjukkan performa sebagai berikut:

- Untuk kelas "LemonFish-4cm", model mencapai *precision* sebesar 94%, *recall* sebesar 92%, *F1-score* sebesar 91% dan AP sebesar 87%.
- Untuk kelas "ManFish-6cm", model memiliki *precision* sebesar 95%, *recall* sebesar 94%, *F1-score* sebesar 94%, dan AP sebesar 89%.
- Secara keseluruhan model memperoleh *mean Average Precision* (mAP) sebesar 57%.

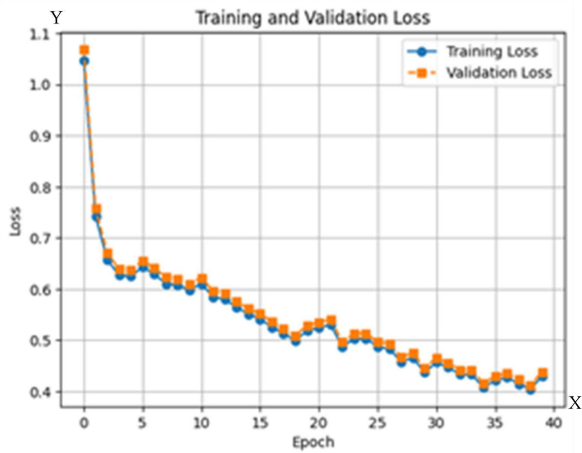
Hasil ini menunjukkan bahwa model memiliki tingkat presisi dan daya tangkap yang cukup tinggi dalam mendeteksi kedua kelas ikan, meskipun masih terdapat ruang untuk peningkatan dalam aspek mAP.

#### 3.2. Training and Validation Loss

Penelitian ini bertujuan untuk membangun dan melatih model deteksi objek menggunakan Faster R-CNN dengan MobileNetV3 sebagai backbone guna mendeteksi ikan pada dataset berformat COCO. Setelah dataset dipersiapkan, model Faster R-CNN diinisialisasi dan dimodifikasi agar sesuai dengan jumlah kelas yang terdapat dalam dataset.

Proses pelatihan dilakukan selama 40 epoch menggunakan optimizer Adam, dengan penyesuaian *momentum* dan *weight decay* untuk mengoptimalkan

kinerja model. Pada setiap epoch, model secara otomatis menghitung nilai *loss*, yang kemudian ditampilkan untuk memantau performa pelatihan serta memastikan bahwa model belajar secara optimal.



Gambar 7. Training and Validation Loss Graphic

```
Epoch 30, Loss: 0.456667
Epoch 30, Training Loss: 0.456667, Validation Loss: 0.465801
100%|██████████| 47/47 [00:07<00:00, 6.15it/s]
Epoch 31, Loss: 0.446476
Epoch 31, Training Loss: 0.446476, Validation Loss: 0.455406
100%|██████████| 47/47 [00:08<00:00, 5.50it/s]
Epoch 32, Loss: 0.433285
Epoch 32, Training Loss: 0.433285, Validation Loss: 0.441951
100%|██████████| 47/47 [00:07<00:00, 5.95it/s]
Epoch 33, Loss: 0.432442
Epoch 33, Training Loss: 0.432442, Validation Loss: 0.441091
100%|██████████| 47/47 [00:08<00:00, 5.77it/s]
Epoch 34, Loss: 0.407600
Epoch 34, Training Loss: 0.407600, Validation Loss: 0.415752
100%|██████████| 47/47 [00:08<00:00, 5.65it/s]
Epoch 35, Loss: 0.421118
Epoch 35, Training Loss: 0.421118, Validation Loss: 0.429541
100%|██████████| 47/47 [00:07<00:00, 6.19it/s]
Epoch 36, Loss: 0.426987
Epoch 36, Training Loss: 0.426987, Validation Loss: 0.435527
100%|██████████| 47/47 [00:08<00:00, 5.54it/s]
Epoch 37, Loss: 0.414444
Epoch 37, Training Loss: 0.414444, Validation Loss: 0.422732
100%|██████████| 47/47 [00:07<00:00, 6.16it/s]
Epoch 38, Loss: 0.403845
Epoch 38, Training Loss: 0.403845, Validation Loss: 0.411922
100%|██████████| 47/47 [00:08<00:00, 5.53it/s]
Epoch 39, Loss: 0.429646
Epoch 39, Training Loss: 0.429646, Validation Loss: 0.438239
```

Gambar 8. Nilai Loss

Berdasarkan **Gambar 8**, dapat diketahui bahwa proses training dilakukan hingga nilai *loss* konsisten berada di bawah 0,5, yang menunjukkan bahwa model telah mencapai tingkat konvergensi yang baik. Dengan nilai *loss* yang rendah, model mampu menghasilkan akurasi yang cukup tinggi dalam mendeteksi objek. Pencapaian ini disebabkan oleh pelatihan yang dilakukan selama 40 epoch, sehingga model memiliki cukup waktu untuk mempelajari pola dalam dataset secara optimal.

### 3.3. Evaluasi Model

Untuk mengevaluasi performa model Faster R-CNN, dilakukan pengujian menggunakan beberapa metrik evaluasi, yaitu *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), dan *R-squared* ( $R^2$ ). Metrik-

metrik ini digunakan untuk mengukur tingkat kesalahan serta keakuratan model dalam mendeteksi objek.

Nilai MSE menunjukkan rata-rata kesalahan kuadrat dalam prediksi, sementara RMSE memberikan gambaran lebih jelas mengenai seberapa besar deviasi hasil prediksi terhadap nilai sebenarnya. Selain itu, *R-squared* ( $R^2$ ) digunakan untuk mengukur tingkat kemampuan model dalam menjelaskan variabilitas data pada pendeteksian objek. Dengan menggunakan metrik ini, dapat diketahui seberapa baik model telah dilatih dan seberapa akurat prediksinya dalam mendeteksi ikan.

Tabel 3. Evaluasi Model

Epoch	MSE	RMSE	MAE	$R^2$
0	0.015978	0.1679	0.1255	0.01121
1	0.015961	0.1676	0.1258	0.01092
2	0.015995	0.1682	0.1256	0.01105
3	0.015982	0.16785	0.1254	0.01076
4	0.01597	0.16765	0.1257	0.01108
5	0.015989	0.1681	0.1259	0.01097
6	0.01596	0.1675	0.1253	0.0111
7	0.01598	0.16795	0.1251	0.01091
8	0.015965	0.1674	0.1258	0.01114
8	0.01599	0.168	0.1257	0.01089
9	0.015978	0.1679	0.1255	0.01121
		...		
31	0.015957	0.16732	0.1251	0.01113
32	0.015992	0.16812	0.12575	0.01085
33	0.015968	0.16758	0.12565	0.01104
34	0.015986	0.16802	0.1258	0.0109
35	0.015953	0.16715	0.1256	0.01117
36	0.015978	0.16782	0.1254	0.01099
37	0.015956	0.16728	0.1252	0.01109
38	0.015994	0.1682	0.1259	0.01083
39	0.015964	0.16746	0.1257	0.011

Berdasarkan hasil pelatihan model yang ditampilkan pada **Tabel 3**, diketahui bahwa model Faster R-CNN memiliki nilai *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE), dan *R-squared* ( $R^2$ ) yang nilai seluruhnya berada di bawah 0,2 setelah melalui 40 epoch pelatihan.

Hasil pengujian ini menunjukkan bahwa model Faster R-CNN mampu melakukan deteksi ikan dengan performa yang baik. Hal ini ditunjukkan oleh nilai MSE, RMSE, dan MAE yang relatif rendah serta nilai  $R^2$  yang konsisten, yang menunjukkan bahwa model mampu mengenali pola dan tren pergerakan ikan secara kompleks dengan tingkat akurasi yang cukup tinggi.

Model tidak hanya mampu mendeteksi ikan secara statis, tetapi juga dapat mempelajari variasi posisi, bentuk, serta arah pergerakan ikan yang kompleks dan dinamis dalam lingkungan akuatik. Meskipun nilai  $R^2$  belum mencapai angka mendekati 1, yang menunjukkan prediksi sempurna, konsistensi nilai-nilai evaluasi seperti MSE, RMSE, dan MAE yang rendah mengindikasikan bahwa prediksi model cukup akurat dengan tingkat kesalahan yang relatif kecil. Hal ini menunjukkan bahwa performa model tergolong layak dan cukup baik untuk diterapkan dalam aplikasi nyata.

#### 4. Kesimpulan

Penelitian ini berhasil mengembangkan model deteksi ikan berbasis *deep learning* dengan menggunakan Faster R-CNN. Model dilatih dengan dataset gambar yang diekstraksi dari video menggunakan Roboflow dan diuji untuk mendeteksi dua jenis ikan, yaitu Manfish dan Lemonfish.

Hasil evaluasi menunjukkan bahwa model mencapai *precision* sebesar 94% untuk Lemonfish dan 95% untuk Manfish, dengan nilai *mean Average Precision* (mAP) sebesar 57%. Meskipun model menunjukkan performa yang baik dalam mendeteksi ikan, masih terdapat beberapa keterbatasan, seperti nilai mAP yang relatif rendah dibandingkan dengan nilai *precision*, *recall*, dan *F1-score*.

Oleh karena itu, penelitian lebih lanjut diperlukan untuk meningkatkan performa model, misalnya dengan menyesuaikan parameter pelatihan, meningkatkan jumlah dataset, atau mengimplementasikan teknik augmentasi data untuk meningkatkan performa deteksi dalam kondisi pencahayaan dan latar belakang yang lebih kompleks. Serta pertimbangkan pemilihan backbone yang memiliki kecepatan dan akurasi tinggi, seperti ResNet atau Inception, untuk mengoptimalkan kinerja model.

#### Daftar Rujukan

- [1] R. F. Syreen and K. Merrilance, "A Survey on Underwater Fish Species Detection and Classification International Journal of Computer Sciences and Engineering Open Access A Survey on Underwater Fish Species Detection and Classification," no. April 2019, 2020, doi: 10.26438/ijcse/v7si8.9598.
- [2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," p. 9, 2015.
- [3] A. Salman *et al.*, "Fish species classification in unconstrained underwater environments based on deep learning," *Limnol. Oceanogr. Methods*, vol. 14, no. 9, pp. 570–585, 2016, doi: 10.1002/lom3.10113.
- [4] M. J. Islam, Y. Xia, and J. Sattar, "Fast Underwater Image Enhancement for Improved Visual Perception," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3227–3234, 2020, doi: 10.1109/LRA.2020.2974710.
- [5] A. Howard, W. Wang, G. Chu, L. Chen, B. Chen, and M. Tan, "Searching for MobileNetV3 Accuracy vs MADDs vs model size," *Int. Conf. Comput. Vis.*, pp. 1314–1324, 2019.
- [6] A. Salman *et al.*, "Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system," *ICES J. Mar. Sci.*, vol. 77, no. 4, pp. 1295–1307, 2020, doi: 10.1093/icesjms/fsz025.
- [7] H. Kim, "Deep Learning," *Artif. Intell. 6G*, vol. 22, no. 4, pp. 247–303, 2022, doi: 10.1007/978-3-030-95041-5\_6.
- [8] L. Liu *et al.*, "Deep Learning for Generic Object Detection: A Survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, 2020, doi: 10.1007/s11263-019-01247-4.
- [9] J. Huang, V. Rathod, and C. Sun, "Speed/accuracy trade-offs for modern convolutional object detectors," *Proc. IEEE CVPR*, 2017.
- [10] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "RefinementNet," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 4203–4212, 2018.
- [11] C.-Y. Fu, M. Shvets, and A. C. Berg, "RetinaMask: Learning to predict masks improves state-of-the-art single-shot detection for free," 2019, [Online]. Available: <http://arxiv.org/abs/1901.03353>
- [12] N. Ghatwary, X. Ye, and M. Zolgharni, "Esophageal Abnormality Detection Using DenseNet Based Faster R-CNN with Gabor Features," *IEEE Access*, vol. 7, pp. 84374–84385, 2019, doi: 10.1109/ACCESS.2019.2925585.
- [13] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.
- [14] R. Girshick, "Fast R-CNN," 2015.
- [15] A. Tharwat, "Classification assessment methods," *Appl. Comput. Informatics*, vol. 17, no. 1, pp. 168–192, 2018, doi: 10.1016/j.aci.2018.08.003.
- [16] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, pp. 1–13, 2020, doi: 10.1186/s12864-019-6413-7.
- [17] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature," *Geosci. Model Dev.*, vol. 7, no. 3, pp. 1247–1250, 2014, doi: 10.5194/gmd-7-1247-2014.
- [18] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Clim. Res.*, vol. 30, no. 1, pp. 79–82, 2005, doi: 10.3354/cr030079.
- [19] D. L. J. Alexander, A. Tropsha, and David A. Winkler, "Beware of R2: simple, unambiguous assessment of the prediction accuracy of QSAR and QSPR models," *Chem. Inf. Model.*, vol. 55, no. 7, pp. 1316–1322, 2015.
- [20] A. Gosiewska, K. Woźnica, and P. Biecek, "Interpretable meta-score for model performance," *Nat. Mach. Intell.*, vol. 4, no. 9, pp. 792–800, 2022, doi: 10.1038/s42256-022-00531-2.
- [21] C. Miller, T. Portlock, D. M. Nyaga, and J. M. O'Sullivan, "A review of model evaluation metrics for machine learning in genetics and genomics," *Front. Bioinform.*, vol. 4, no. September, pp. 1–13, 2024, doi: 10.3389/fbinf.2024.1457619.
- [22] D. M. W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," pp. 37–63, 2020, [Online]. Available: <http://arxiv.org/abs/2010.16061>

