

RESEARCH ARTICLE

Perancangan dan Implementasi *Deep Learning* untuk Deteksi Kesegaran Ikan Beku pada Aplikasi FishQ menggunakan YOLOv8

Aldra Kasyfil Aziz , Ledya Novamizanti* and Suryo Adhi Wibowo

Fakultas Teknik Elektro, Universitas Telkom, Bandung, 40257, Jawa Barat, Indonesia

*Corresponding author: ledyaldn@telkomuniversity.ac.id

Abstrak

Industri perikanan di Indonesia memegang peran vital dalam perekonomian nasional dan kesejahteraan masyarakat, namun proses sortasi ikan yang masih sering dilakukan secara konvensional dan mengandalkan pengamatan visual rentan terhadap kesalahan dan kurang efisien. Untuk meningkatkan akurasi dan efisiensi dalam penilaian kesegaran ikan, dikembangkanlah FishQ, sebuah aplikasi berbasis *deep learning* dengan model YOLOv8 yang mampu mendeteksi dan mengklasifikasikan kesegaran ikan secara otomatis. FishQ dirancang untuk diakses oleh berbagai pemangku kepentingan dalam rantai pasok perikanan dan telah menunjukkan hasil penelitian yang memuaskan, dengan kemampuan mengidentifikasi kesegaran ikan cakalang dalam kondisi beku dengan nilai mAP 94,7% serta mendeteksi hingga 94% citra ikan secara tepat. Penggunaan teknologi ini diharapkan dapat mempercepat proses penilaian kesegaran ikan dibandingkan dengan metode konvensional, sehingga dapat meningkatkan kualitas produk dan mengurangi kerugian ekonomi.

Key words: FishQ, Kesegaran Ikan, *Deep Learning*, YOLOv8, Deteksi Objek

Pendahuluan

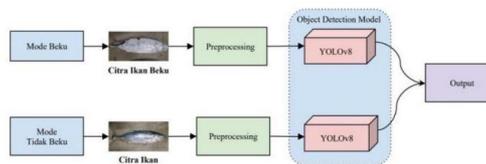
Indonesia memiliki luas perairan 6.400.000 km² dari total wilayah 8.300.000 km² [1], menjadikannya unggul di sektor perikanan. Produk perikanan Indonesia, termasuk 52.053 ton ikan segar yang diekspor pada tahun 2022 [2], tidak hanya memenuhi pasar dalam negeri tetapi juga mancanegara. Ikan TCT (Tuna Cakalang Tongkol) merupakan andalan ekonomi dengan kontribusi 20,06 terhadap produksi global pada 2019 [3]. Kualitas komoditas laut sangat bergantung pada teknik penanganannya, yang meliputi penangkapan, sortasi, pencucian, penirisan, dan penyimpanan ikan [4]. Sortasi penting untuk menjaga kualitas dan menentukan ikan yang memenuhi standar ekspor. Namun, sortasi konvensional sulit dan memakan waktu, terutama untuk ikan beku. Aruna, perusahaan ekonomi kelautan dan perikanan, saat ini menggunakan metode sortasi konvensional yang tidak efisien. Oleh karena itu, diperlukan inovasi dengan teknologi digital dan metode *deep learning* untuk mengidentifikasi kualitas ikan secara lebih efisien.

Tinjauan Pustaka

Pada bagian ini berisi kajian teori yang terkait dengan penelitian. Beberapa teori yang akan dijelaskan yaitu *Deep Learning*, *Object Detection*, Data Mentah Citra Ikan, *Preprocessing*.

Deep Learning

Deep learning merupakan pengembangan dari neural networks tradisional yang menunjukkan peningkatan signifikan dari akarnya. Selain



Gambar 1. Blok Diagram *Deep Learning*.

itu, *deep learning* secara bersamaan memanfaatkan transformasi dan teknologi grafik untuk membangun model pembelajaran *multi-layer*. Teknik *deep learning* yang paling baru dikembangkan telah mencapai kinerja yang sangat baik di berbagai aplikasi, termasuk pemrosesan audio dan ucapan, pemrosesan data visual, NLP (*Natural Language Processing*), dan lain-lain [5], [6], [7], [8]. Gambar 1 menunjukkan blok diagram *deep learning* yang akan digunakan pada aplikasi *mobile* dengan deteksi objek secara *non real-time*.

Gambar 1 adalah blok diagram *deep learning* yang menunjukkan proses identifikasi kesegaran ikan. Ada dua mode *input* dalam sistem ini, yaitu "mode beku" dan "mode tidak beku". Mode beku menerima citra ikan beku sebagai *input*, sedangkan mode tidak beku menerima citra ikan tidak beku. Setelah citra diterima, selanjutnya diproses melalui tahap *preprocessing*. Tahap ini melibatkan proses normalisasi dan *resize* citra untuk mempersiapkan citra sebelum proses analisis lebih lanjut. Citra yang telah diproses kemudian dilanjutkan ke model deteksi

Table 1. Data Mentah Citra Ikan

Jenis	Kesegaran	Jumlah Ikan	Jumlah
	<i>fresh</i>	= 1	600
Beku	<i>no-fresh</i>	= 1	600
	<i>multiple</i>	> 1	600
Total			3.600

Table 2. Statistik *Bit Avalanche Test*

No.	Model	mAP	Waktu Proses Deteksi (s)
1.	YOLOv8	0,695	0,394
2.	YOLOv7	0,048	1,195
3.	YOLOv6	0,211	0,390
4.	DETR	0,638	4,161

objek YOLOv8. Hasil dari model deteksi objek kemudian ditampilkan sebagai output berupa label ikan "*fresh*" atau "*no-fresh*".

Data Mentah Citra Ikan

Dataset untuk model *deep learning* ini terdiri dari enam kategori ikan dengan total 3.600 citra. Dataset ini digunakan untuk melatih model *deep learning* untuk mengenali dan membedakan dua kondisi ikan yang diharapkan dapat meningkatkan akurasi model. Tabel 1 menunjukkan distribusi dataset citra ikan yang dibagi menjadi tiga bagian yaitu *train*, *validation*, dan *test*. Dataset *train* yang terdiri dari 420 citra untuk setiap kategori ikan digunakan untuk melatih model. Selama proses pelatihan, model mempelajari fitur-fitur dari citra. Selanjutnya, dataset *validation* yang mencakup 120 citra untuk setiap kategori digunakan untuk mengatur parameter model dan memvalidasi *performanya* selama proses pelatihan. Proses validasi membantu dalam mencegah *overfitting*, yaitu ketika model terlalu spesifik untuk data pelatihan dan *performanya* menurun pada data baru. Validasi memberikan *feedback* selama proses pelatihan dan digunakan untuk mengatur hiperparameter seperti *learning rate*. Terakhir, dataset *test* yang mencakup 60 citra untuk setiap kategori digunakan untuk menguji *performa* model setelah pelatihan selesai. Proses *testing* memberikan gambaran mengenai *performa* model pada data yang belum pernah dilihat sebelumnya.

Preprocessing

Pada tahap *preprocessing*, citra ikan dioptimalkan dan dimodifikasi untuk memaksimalkan efisiensi pemrosesan oleh model *deep learning*. Proses modifikasi mencakup normalisasi nilai piksel, mengatur resolusi, dan augmentasi. Augmentasi yang digunakan adalah *flip*, *rotation*, *saturation*, *brightness*, dan *exposure* sehingga menambah jumlah data dengan geometri yang berbeda. Penambahan citra oleh augmentasi membuat jumlah data menjadi tiga kali data awal, sehingga setelah augmentasi terdapat 8.640 citra.

Object Detection

Object detection merupakan proses untuk mendeteksi keberadaan objek pada sebuah citra atau video dan mengklasifikasikannya ke dalam kelas objek tertentu. Dalam aplikasi *mobile* deteksi ikan ini, *object detection* digunakan untuk mendeteksi keberadaan ikan pada citra yang diinputkan pengguna. Penelitian ini melakukan percobaan terhadap beberapa model *object detection* yaitu YOLOv8, YOLOv7, YOLOv6, dan DETR untuk menentukan model terbaik yang akan digunakan dalam dataset ikan cakalang. Tabel 2 menunjukkan bahwa

model YOLOv8 memiliki nilai mAP (*mean Average Precision*) tertinggi yaitu 0,695 dibandingkan dengan model lainnya. Hal ini menunjukkan bahwa YOLOv8 memiliki akurasi deteksi yang lebih baik dengan menghasilkan prediksi yang lebih tepat dalam mendeteksi objek. Selain itu, waktu proses deteksi YOLOv8 juga relatif cepat, yaitu 0,394 detik per gambar, menjadikannya efisien dalam hal kecepatan. YOLOv8 merupakan pengembangan terbaru dari seri algoritma YOLO (You Only Look Once) untuk *object detection*. YOLOv8 menggabungkan berbagai keunggulan dari model pendahulunya seperti backbone CSPNet (Cross Stage Partial Network) yang kuat, SC-Net (Spatial Channel Attention), MS-FFNet (Multi-Scale Feature Fusion), optimizer AdamW, serta memanfaatkan data augmentasi dan hyperparameter tuning.

Cara kerja YOLOv8 dimulai dengan mengambil citra dan menggunakan CSPNet untuk mengekstrak fitur dari citra tersebut. Kemudian, fitur-fitur ini melewati proses yang disebut *Feature Pyramid Network* (FPN) untuk mengekstrak fitur pada berbagai ukuran. Fitur-fitur ini kemudian digabungkan menggunakan MS-FFNet. Selanjutnya, algoritma ini menggunakan SC-Net untuk fokus pada bagian dan fitur citra yang paling penting. Terakhir, fitur-fitur ini dilewatkan ke *classifier* dan *regressor* untuk mendeteksi objek. Dengan menerapkan YOLOv8, diharapkan dapat mendeteksi ikan pada citra *input* secara akurat dan membatasi ikan tersebut dalam *bounding box*. Hasil deteksi ini selanjutnya digunakan untuk memisahkan ikan dari latar belakang sehingga dapat diproses lebih lanjut untuk klasifikasi kesegarannya.

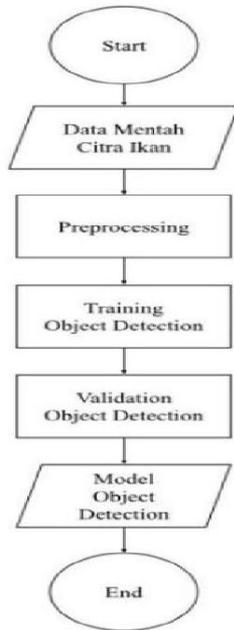
Proses pelatihan model *object detection* diawali dengan persiapan data berupa citra ikan dalam keadaan beku dan tidak beku, baik yang segar maupun tidak segar. Data mentah citra ikan ini kemudian dilakukan *preprocessing* serta labeling objek ikan pada setiap citra. Langkah berikutnya adalah pelatihan model *object detection* menggunakan algoritma YOLOv8. Sebelumnya model YOLOv8 telah dilatih pada dataset COCO. Oleh karena itu, dilakukan teknik transfer *learning* dengan cara *fine-tuning* model YOLOv8 yang sudah terlatih tersebut pada dataset citra ikan. Proses finetuning melibatkan pengaturan *hyperparameter* seperti *learning rate*, *batch size*, dan *epoch*. Setelah proses *training* selesai, dilakukan evaluasi kinerja model menggunakan data validasi dengan metrik seperti *precision*, *accuracy*, *recall*, *F1-score*, *mAP*, dan lain-lain. Model *object detection* yang telah dilatih dan dievaluasi dengan baik ini kemudian siap digunakan untuk mendeteksi ikan pada *input* citra. Jika model sudah tervalidasi, maka dihasilkan model YOLOv8 yang dapat digunakan untuk deteksi objek ikan pada citra baru. Gambar 2 menunjukkan proses pelatihan model *object detection*.

Metodologi Penelitian

Implementasi dari *FishQ*: Aplikasi Pengidentifikasi Kesegaran Ikan menggunakan Metode *Deep Learning* merupakan proses pengembangan aplikasi *mobile* yang berfokus pada kemudahan penggunaan dan akurasi dalam mengidentifikasi kesegaran ikan cakalang. Aplikasi ini ditujukan untuk membantu perusahaan perikanan, khususnya perusahaan Aruna dalam meningkatkan kecepatan dan akurasi proses sortasi kesegaran ikan cakalang. Implementasi ini melibatkan serangkaian tahapan pengembangan perangkat lunak yang diintegrasikan dengan model *deep learning*. Proses pengembangan model *deep learning* terdiri dari tiga tahap utama yaitu akuisisi data, *preprocessing*, pelatihan model, dan pengujian *deep learning*.

Akuisisi Data

Pada tahap pengembangan model, langkah pertama adalah akuisisi data dengan mengumpulkan citra ikan dari perusahaan Aruna. Selain itu, penelitian ini juga mengumpulkan data dari internet dan membuat data sintesis untuk memperkaya variasi dan jumlah citra yang tersedia. Dataset sintesis adalah kumpulan data yang dibuat untuk melatih



Gambar 2. Flowchart Proses Pelatihan Model *Object Detection*.

Table 3. Statistik *Bit Avalanche Test*

No.	Model	mAP	Waktu Proses Deteksi (s)
1.	YOLOv8	0,695	0,394
2.	YOLOv7	0,048	1,195
3.	YOLOv6	0,211	0,390
4.	DETR	0,638	4,161

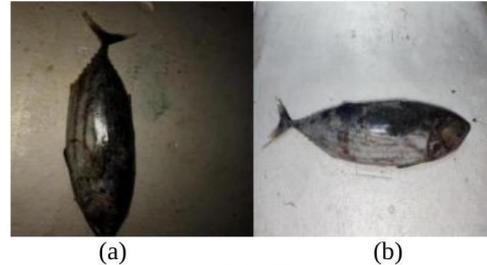
model ketika data ril sulit diperoleh dan menghasilkan citra yang berbeda posisi dan latar belakang [9]. Data yang terkumpul terdiri dari dua kondisi ikan, yaitu ikan beku sebanyak 1.887 citra dan ikan tidak beku sebanyak 1.361 citra yang digunakan untuk melatih model YOLOv8. Rincian dataset dapat dilihat pada Tabel 3.

Preprocessing

Preprocessing data menggunakan platform *Roboflow* dilakukan untuk memperkaya dan meningkatkan keragaman data yang dapat digunakan oleh model dalam proses pelatihan [10]. *Roboflow* adalah salah satu platform alat anotasi citra yang menggunakan fitur seperti *bounding box* dan *object labeling*, serta alat [Opt] *preprocessing* dan augmentasi citra untuk membuat dataset yang lebih bervariasi [11]. Dataset yang telah terkumpul akan digunakan dalam melatih model *deep learning* untuk mengenali dan membedakan dua kondisi ikan yaitu "fresh" dan "nofresh". Sebelum digunakan untuk melatih model, citra masuk ke tahap *preprocessing* dan augmentation. Pada tahap *preprocessing*, normalisasi dan ukuran citra diatur dengan ukuran yang sama yaitu 640×640 piksel. Sedangkan pada tahap augmentasi dilakukan proses memperbanyak jumlah data yang awalnya berjumlah 1.887 citra menjadi 4.531 citra dengan pembagian untuk masing-masing dataset adalah *train set* 88%, *valid set* 8%, dan *test set* 4%. Augmentasi yang digunakan adalah flip (horizontal & vertical), rotation (between -15° & 15°), saturation (between -25% & +25%), brightness (between -15% & +15%), dan *exposure* (between -10% & +10%). Kemudian data hasil augmentasi digunakan untuk melatih model *deep learning*



Gambar 3. Sampel Dataset (a) Kelas "Fresh" (b) Kelas "No-Fresh"



Gambar 4. Sampel Dataset Hasil Augmentasi (a) Kelas "Fresh" (b) Kelas "NoFresh"

YOLOv8. Sampel citra dari dataset ikan cakalang terlihat pada Gambar 3, gambar (a) merupakan contoh citra ikan untuk kelas "fresh", sedangkan gambar (b) adalah untuk kelas "no-fresh". Sampel citra hasil augmentasi terlihat pada Gambar 4, gambar (a) merupakan contoh citra ikan untuk kelas "fresh", sedangkan gambar (b) adalah untuk kelas "nofresh". *Object Detection* merupakan teknologi komputer terkait dengan visi komputer dan pemrosesan citra yang berfungsi untuk mendeteksi keberadaan objek-objek semantik dari kategori tertentu dalam citra digital maupun video [12]. Dalam aplikasi *mobile* deteksi ikan ini, *object detection* digunakan untuk mendeteksi keberadaan ikan pada citra yang diinputkan pengguna.

Algoritma *object detection* yang digunakan adalah YOLOv8. YOLO merupakan algoritma yang dirancang untuk mendeteksi objek. Cara kerjanya adalah dengan membagi citra *input* menjadi beberapa kotak grid, kemudian memprediksi *bounding box* (kotak pembatas) beserta probabilitas untuk setiap kotak tersebut. Jika pusat suatu objek terletak di dalam salah satu kotak grid, maka kotak tersebut akan mendeteksi keberadaan objek. Setiap kotak grid memprediksi *bounding box* dan *confidence score* (skor kepercayaan) untuk setiap *bounding box* yang diprediksi. *Confidence score* merepresentasikan tingkat keyakinan dan akurasi model bahwa terdapat objek di dalam *bounding box* tersebut dalam bentuk persentase [13]. Dengan pengimplementasian YOLOv8, diharapkan aplikasi dapat mendeteksi ikan pada citra *input* secara akurat dan membatasi ikan tersebut dalam *bounding box*.

Pelatihan Model

Pelatihan model YOLOv8 dilakukan dengan memanfaatkan Google Colab Pro. Google Colab merupakan IDE (*Integrated Development Environment*) yang memungkinkan pengguna untuk menulis dan menjalankan kode Python, di mana proses komputasinya dieksekusi oleh server milik Google yang dilengkapi dengan perangkat keras berkinerja tinggi [14]. Python dikenal sebagai bahasa pemrograman yang menawarkan kombinasi kemampuan dan sintaksis kode yang sederhana serta mudah dipahami. Kelebihannya juga terletak pada koleksi *library* standar yang luas dan lengkap sehingga memungkinkan pengembang untuk mengakses berbagai fungsi dan utilitas secara mudah. Dengan kelebihan tersebut, Python dinilai sebagai bahasa pemrograman yang handal dan fleksibel untuk berbagai keperluan pengembangan aplikasi [15]. Python sering dimanfaatkan sebagai bahasa *scripting*,

namun dalam praktiknya, penggunaan Python jauh lebih luas dan menjangkau berbagai konteks yang biasanya tidak ditangani oleh bahasa *scripting* pada umumnya [16]. Bagian kode ini melakukan perintah untuk memulai proses pelatihan model deteksi objek menggunakan *library Ultralytics*. Pertama, kode menjalankan perintah "yolo" dengan beberapa parameter konfigurasi untuk melatih model deteksi objek.

Parameter "task=detect" menentukan tugas yang akan dilakukan, yaitu deteksi objek, dan "mode=train" menentukan mode operasi sebagai pelatihan model. Kode menggunakan model YOLOv81 ("yolov81.pt") dan menentukan lokasi file konfigurasi data ("data.yaml") yang terdapat di dalam direktori dataset yang telah diunduh sebelumnya. Selanjutnya, kode mengatur jumlah *epoch* (iterasi pelatihan) sebanyak 200 *epoch* dan ukuran gambar *input* sebesar 640 x 640 piksel. Parameter "plots=True" akan menghasilkan plot visualisasi selama proses pelatihan berlangsung. Kombinasi "optimizer='AdamW'", "lr0 = 0, 01", dan "momentum= 0,9" digunakan untuk meningkatkan efisiensi dan stabilitas pelatihan model *deep learning*.

Pengujian Deep Learning

Pengujian spesifikasi *deep learning* merupakan tahapan awal dalam pengembangan aplikasi *FishQ* untuk memastikan model YOLOv8 yang digunakan dapat mengidentifikasi kesegaran ikan dengan tingkat akurasi yang tinggi. Skenario pengujian ini dirancang untuk mengevaluasi kemampuan model dalam membedakan ikan segar dan tidak segar, baik dalam kondisi beku maupun tidak beku. Proses pengujian akan melibatkan serangkaian langkah yang terstruktur dan sistematis. Pertama, dataset citra ikan akan dibagi menjadi data *training*, *validation*, dan *test* [17]. Data *training* akan digunakan untuk melatih model YOLOv8, sementara data *validation* akan digunakan untuk menyesuaikan hyperparameter dan mengevaluasi *performa* model selama proses pelatihan. Setelah itu, data *test* akan digunakan untuk menilai kemampuan model dalam mengidentifikasi kesegaran ikan pada data yang belum pernah dilihat sebelumnya [18]. Selama proses pengujian, berbagai metrik evaluasi akan digunakan untuk mengukur *performa* model yaitu *accuracy*, *precision*, *recall*, *specificity*, dan *F1-score*. Metrik-metrik tersebut akan memberikan informasi mengenai seberapa baik model dapat mendeteksi ikan segar dan tidak segar dengan benar, serta seberapa konsisten model dalam melakukan prediksi [19]. Hasil pengujian akan menjadi landasan untuk memastikan keandalan dan efektivitas aplikasi dalam membantu industri perikanan meningkatkan efisiensi dan kualitas proses sortasi ikan.

Hasil dan Pembahasan

Hasil Pengujian Deep Learning

Model *deep learning* yang sudah dilatih selanjutnya diujikan terhadap data validasi. Hasil pengujian divisualisasikan dalam bentuk *confusion matrix* seperti pada Gambar 5. Berdasarkan *confusion matrix* berikut didapatkan nilai TP sebesar 754 untuk ikan "fresh" dan 653 untuk ikan "no-fresh" dengan FP yang didapatkan sebesar 110 untuk ikan "fresh" dan 103 untuk ikan "no-fresh". Sedangkan TN sebesar 687 untuk ikan "fresh" dan 827 untuk ikan "no-fresh" dengan FN sebesar 74 untuk ikan "fresh" dan 42 untuk ikan "no-fresh". Gambar 6 merupakan serangkaian grafik hasil pelatihan model. Berikut adalah penjelasan metrik proses latih model dalam citra tersebut.

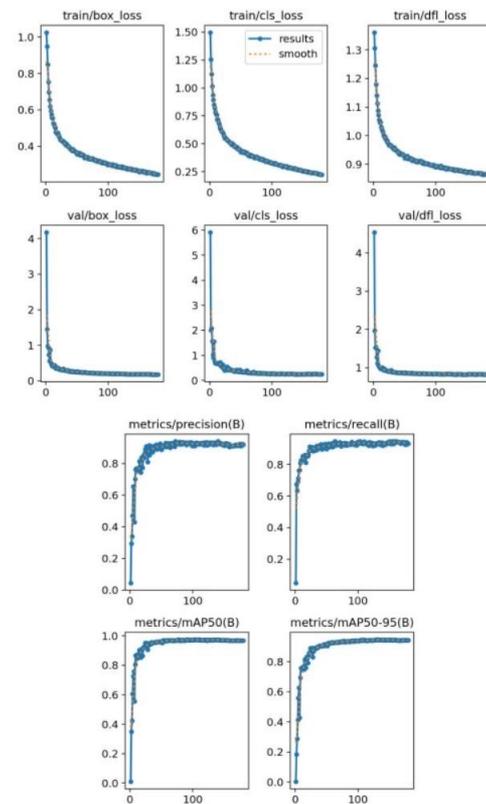
- **Training Loss**

Dalam proses pelatihan model, grafik "train/box_loss" menunjukkan penurunan nilai box loss yang signifikan dari 1,0 hingga 0,4 dan semakin turun seiring berjalannya waktu dalam 180 *epoch*. Hal ini mengindikasikan bahwa model semakin akurat dalam memprediksi lokasi objek. Grafik

"train/cls_loss" menggambarkan penurunan nilai *classification loss*

PREDICTED	ACTUAL		
	Fresh	No-Fresh	Background
Fresh	754	42	68
No-Fresh	69	653	34
Background	5	0	0

Gambar 5. Confusion Matrix YOLOv8



Gambar 6. Sampel Dataset Hasil Augmentasi (a) Kelas "Fresh" (b) Kelas "NoFresh"

yang signifikan dari 1,5 hingga 0,5 dan semakin turun sepanjang iterasi yang menandakan peningkatan kemampuan model dalam mengklasifikasikan objek. Sementara itu, grafik "train/df_l_loss" menampilkan penurunan nilai loss distribusi fitur lokasi yang signifikan dari 1,3 hingga 0,9 dan semakin turun sepanjang iterasi yang menunjukkan peningkatan dalam memprediksi fitur-fitur spesifik dari objek yang terdeteksi.

- **Performa Model**

Table 4. Hasil Pelatihan Model *Deep Learning*

Parameter	YOLOv8 Model
<i>mAP</i>	0,947
<i>Accuracy</i>	0,899
<i>Precision</i>	0,868
<i>Recall</i>	0,925
<i>Specificity</i>	0,876
<i>F1-Score</i>	0,896
<i>Loss (Box)</i>	0,246
<i>Loss (DFL)</i>	0,865
Waktu deteksi/citra	0,124

Grafik "metrics/precision(B)" dan "metrics/recall(B)" keduanya menunjukkan peningkatan *performa* model. *Precision*, yang mengukur proporsi prediksi positif yang benar meningkat mendekati nilai 1,0, menunjukkan tingkat kesalahan yang rendah dalam prediksi positif. *Recall*, yang mengukur kemampuan model dalam menemukan semua kasus positif meningkat mendekati nilai 1,0, menandakan bahwa model dapat mendeteksi semua objek yang relevan dengan baik.

- Validasi Model

Pada set validasi, grafik "val/box loss" menunjukkan penurunan nilai loss box dari sekitar 4,0 hingga kurang dari 1,0, menunjukkan bahwa model *berperforma* baik pada data yang tidak dilihat selama pelatihan. Grafik "val/cls loss" dan "val/df.Loss" juga menunjukkan penurunan nilai yang signifikan, menandakan peningkatan kemampuan model dalam mengklasifikasikan objek dan memprediksi fitur-fitur spesifik pada data validasi.

- Akurasi Model

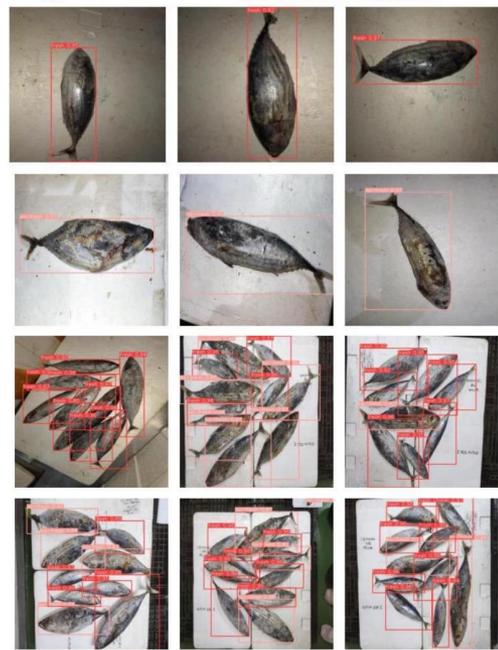
Grafik "metrics/mAP@0,50" dan "metrics/mAP@0,50:0,95" menampilkan mAP (*mean Average Precision*) model pada berbagai ambang batas IoU (*Intersection over Union*). Nilai mAP meningkat secara signifikan dari 0,2 hingga mendekati 1,0 untuk IoU 0,50, dan dari sekitar 0,2 hingga mendekati 1,0 untuk rentang IoU 0,50 hingga 0,95 dalam 180 *epoch*. Peningkatan ini menunjukkan peningkatan akurasi model dalam mendeteksi dan mengklasifikasikan objek dengan berbagai tingkat ketepatan.

Secara keseluruhan, grafik-grafik tersebut menunjukkan bahwa model *deep learning* yang dilatih menunjukkan peningkatan *performa* yang konsisten sepanjang proses pelatihan, baik pada data pelatihan maupun validasi. Tabel 4 berikut merupakan tabel hasil pelatihan model *deep learning*.

Pada Gambar 7 ditunjukkan beberapa citra hasil pengujian *deep learning* yang telah dilatih dan dievaluasi menggunakan dataset ikan beku.

Analisis Hasil Pengujian *Deep Learning*

Pengujian *deep learning* dilakukan dengan membandingkan berbagai metrik *performa* model. Berdasarkan Tabel 4, dapat dilihat bahwa model memiliki nilai mAP 94,7%. Hal ini menunjukkan bahwa model tersebut akurat dalam mendeteksi kesegaran ikan cakalang dalam kondisi beku. Pengujian lebih lanjut dilakukan dengan menggunakan *confusion matrix* untuk mengevaluasi kinerja model *deep learning* dalam mendeteksi kesegaran ikan. Berdasarkan *confusion matrix* pada

**Gambar 7.** Citra Hasil Deteksi

Gambar 7, didapatkan nilai *accuracy* sebesar 89,9%; *precision* sebesar 86,8%; *recall* sebesar 92,5%; *specificity* sebesar 87,6%; dan *F1-score* sebesar 89,6%.

Grafik *performa* pelatihan dan validasi model ditunjukkan pada Gambar 5. Grafik ini mencakup metrik seperti *train box loss*, *train classification loss*, *train DFL loss*, *val box loss*, *val classification loss*, *val DFL loss*, *precision*, *recall*, *mAP@0.5*, dan *mAP@0.5-0.95*. Model tersebut terdapat penurunan *box loss* dari sekitar 0.4 menjadi kurang dari 0.25, serta *precision* dan *recall* yang stabil di sekitar nilai yang tinggi antara 0,90 hingga 1,00. Ini menunjukkan *performa* model yang baik dalam mendeteksi objek pada citra yang tidak dilihat selama pelatihan. Analisis hasil pengujian menunjukkan bahwa model *deep learning* menunjukkan *performa* yang memadai untuk aplikasi dalam mendeteksi kesegaran ikan, dengan berbagai metrik *performa* yang tinggi dan waktu deteksi yang efisien.

Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan terhadap sistem aplikasi FishQ, dapat disimpulkan bahwa model *deep learning* yang dikembangkan untuk mendeteksi kesegaran ikan cakalang dalam kondisi beku menunjukkan *performa* yang baik. Model tersebut memiliki nilai mAP sebesar 94,7%. Berdasarkan *confusion matrix*, model tersebut memiliki nilai *accuracy* 89,9%, *precision* 86,8%, *recall* 92,5%, *specificity* 87,6%, dan *F1-score* 89,6%. Hal ini menunjukkan *performa* model baik dalam mendeteksi kesegaran ikan cakalang.

Daftar Pustaka

1. Pusat Hidro-Oseanografi TNI Angkatan Laut. Data Kelautan Yang Menjadi Rujukan Nasional Diluncurkan; 2018. Accessed: Oct. 14, 2023. Available from: <https://sidakokkhl.kkp.go.id/sidako/data-kelautan>.
2. Badan Pusat Statistik. Ekspor Ikan Segar/Dingin Hasil Tangkap menurut Negara Tujuan Utama, 2012-2021; 2019. Accessed:

- Oct. 14, 2022. Available from: <https://www.bps.go.id/id/statisticstable/1/MjAyNCMx>.
3. Hartanto TR, Suharno S, Burhanuddin B. Daya Saing Ekspor Ikan Tuna-Cakalang-Tongkol Indonesia di Pasar Amerika Serikat. *Jurnal Pengolahan Hasil Perikanan Indonesia*. 2021 Aug;24(2):227-35.
 4. Litaay C, Wisudo SH, Arfah H. Penanganan Ikan Cakalang oleh Nelayan Pole and Line. *Jurnal Pengolahan Hasil Perikanan Indonesia*. 2020 Apr;23(1):112-21.
 5. Adeel A, Gogate M, Hussain A. Contextual deep learning-based audio-visual switching for speech enhancement in real-world environments. *Information Fusion*. 2020 Jul;59:163-70.
 6. Tian H, Chen SC, Shyu ML. Evolutionary Programming Based Deep Learning Feature Selection and Network Construction for Visual Data Classification. *Information Systems Frontiers*. 2020 Jun;22(5):1053-66.
 7. Young T, Hazarika D, Poria S, Cambria E. Recent Trends in Deep Learning Based Natural Language Processing [Review Article]. *IEEE Computational Intelligence Magazine*. 2018 Aug;13(3):55-75.
 8. Koppe G, Meyer-Lindenberg A, Durstewitz D. Deep learning for small and big data in psychiatry. *Neuropsychopharmacology*. 2020 Jul;46(1):176-90.
 9. Tremblay J, To T, Birchfield S. Falling Things: A Synthetic Dataset for 3D Object Detection and Pose Estimation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); 2018. .
 10. Sanjaya J, Ayub M. Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup. *Jurnal Teknik Informatika dan Sistem Informasi*. 2020 Aug;6(2).
 11. Imran A, Setianingsih C, Saputra RE. Deteksi Pelanggaran Pada Bahu Jalan Tol Dengan Intelligent Transportation System Menggunakan Algoritma Yolov5. *eProceedings of Engineering*. 2023;10(5).
 12. Ardiansyah MR, Supit Y, Said MS. Sistem Visi Komputer untuk Kalkulasi Kepadatan Kendaraan menggunakan Algoritma YOLO. *Simtek: Jurnal Sistem Informasi dan Teknik Komputer*. 2022 April;7(1):52-9.
 13. Sugandi AN, Hartono B. Implementasi Pengolahan Citra pada Quadcopter untuk Deteksi Manusia Menggunakan Algoritma YOLO. In: Proc. Industrial Research Workshop and National Seminar. vol. 13; 2022. p. 183-8.
 14. Adeel A, Gogate M, Hussain A. Contextual Deep Learning-Based Audio-Visual Switching for Speech Enhancement in Real-World Environments. *Information Fusion*. 2020 July;59:163-70.
 15. Tian H, Chen SC, Shyu ML. Evolutionary Programming Based Deep Learning Feature Selection and Network Construction for Visual Data Classification. *Information Systems Frontiers*. 2020 June;22(5):1053-66.
 16. Young T, Hazarika D, Poria S, Cambria E. Recent Trends in Deep Learning Based Natural Language Processing [Review Article]. *IEEE Computational Intelligence Magazine*. 2018 August;13(3):55-75.
 17. Prasmatio RM, Rahmat B, Yuniar I. Deteksi dan Pengenalan Ikan Menggunakan Algoritma Convolutional Neural Network. *Jurnal Informatika dan Sistem Informasi*. 2020;1(2):510-21.
 18. Styorini W, Pratiwi A, Widiasari C. Identifikasi Tingkat Kesegaran Ikan Berbasis Android. *Jurnal Amplifier: Jurnal Ilmiah Bidang Teknik Elektro dan Komputer*. 2022 May;12(1):12-8.
 19. Firdausy N, Yuadi I, Puspitasari I. Analisis Sentimen Evaluasi Reaksi E-Learning Menggunakan Algoritma Naïve Bayes, Support Vector Machine Dan Deep Learning. *TechnoCom*. 2023 August;22(3):677-89.
 20. Lathifah HM, Novamizanti L, Rizal S. Fast and Accurate Fish Classification from Underwater Video Using You Only Look Once. In: IOP Conference Series: Materials Science and Engineering. vol. 982; 2020. p. 012003.
 21. Akhyar F, Novamizanti L, Riantiarni T. Sistem Inspeksi Cacat pada Permukaan Kayu Menggunakan Model Deteksi Obyek YOLOv5. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*. 2022;10(4):990.
 22. Aziz AK, Maulana MD, Adawiyah RF, Firdaus RF, Novamizanti L, Ramdhon F. Comparative Analysis of YOLOv8 Models in Skip-jack Fish Quality Assessment System. In: 2023 3rd International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA); 2023. p. 237-42.
 23. Azizah S, Wahidin, Padang M, Novamizanti L, Saidah S. Identifying the Ripeness and Quality Level of Strawberries Based on YOLOv7-EfficientNet. In: The 5th International Conference on Data Science and Its Applications (ICoDSA); 2024. .
 24. Wicaksono BA, Novamizanti L, Ibrahim N. Tea Leaf Maturity Levels Based on YCbCr Color Space and Clustering Centroid. In: Journal of Physics: Conference Series. vol. 1367; 2019. p. 012028.
 25. Akhyar F, Novamizanti L, Putra T, Furqon EN, Chang MC, Lin CY. Lightning YOLOv4 for a Surface Defect Detection System for Sawn Lumber. In: 2022 IEEE 5th International Conference on Multimedia Information Processing and Retrieval (MIPR); 2022. p. 184-9.