Jurnal Elektronika dan Teknik Informatika Terapan (JENTIK)
Volume 4 Nomor 2 Juni 2026
e-ISSN : 2988-0874; p-ISSN : 2988-0866, Hal 43-59
DOI: https://doi.org/10.59061/jentik.v4i2.1627
Tersedia: https://e-journal.poltek-kampar.ac.id/index.php/JENTIK

Evaluasi Efisiensi Komputasi Model MobileNetV2 Berbasis *TensorFlow Lite* untuk Klasifikasi Kesegaran Tiga Spesies Ikan pada Perangkat Edge

Aisyah Nur Nabila[1*], Teguh Junian Kuswanto[2], Santi Febrianti[3], Nuke Amalia[4], Riyan Bagus Prihandanu[5], Rudy Suryadi[6], Emy Sofia[7]
[1]Politeknik Perkapalan Negeri Surabaya, Indonesia
Alamat: Jl. Teknik Kimia, Keputih, Kec. Sukolilo, Surabaya, Jawa Timur 60111
**Korespondensi penulis: aisyahnur@ppns.ac.id*

Abstract*. Computer vision-based fish freshness classification offers the potential to support rapid and consistent visual assessment. However, deploying deep learning models on edge devices requires consideration of both classification performance and computational efficiency. This study evaluates a MobileNetV2 model implemented with TensorFlow Lite to classify fresh and non-fresh conditions in three fish species: Johnius heterolepis, Rastrelliger brachysoma, and Nemipterus bathybius. The dataset consists of 372 images, including 300 training images and 72 validation images. Classification performance is evaluated using a confusion matrix and the accuracy, precision, recall, and F1-score metrics. Based on the reported results, the model correctly classified 69 of the 72 validation images, achieving an accuracy of 95.83%. The misclassifications indicate that recognition performance varies across classes. The computational efficiency evaluation is designed to consider model file size, inference latency, and throughput across Float32, Float16, and INT8 configurations. However, the efficiency values presented have not been verified through direct measurements on an edge device. Consequently, the configuration offering the best balance between classification accuracy and computational efficiency cannot yet be determined. Further research should conduct device-level benchmarking under consistent testing procedures and use an independent test dataset. This study provides an initial basis for developing image-based fish freshness classification systems, although the findings remain limited by the dataset size and the labeling procedures employed.*

Keywords*: fish freshness classification; MobileNetV2; TensorFlow Lite; computational efficiency; model quantization; edge devices.*

Abstrak. Klasifikasi kesegaran ikan berbasis *computer vision* berpotensi mendukung penilaian visual secara cepat dan konsisten. Namun, implementasi model *deep learning* pada perangkat *edge* perlu mempertimbangkan kinerja klasifikasi dan efisiensi komputasi. Penelitian ini mengevaluasi model MobileNetV2 berbasis *TensorFlow Lite* untuk mengklasifikasikan kondisi segar dan tidak segar pada tiga spesies ikan, yaitu *Johnius heterolepis*, *Rastrelliger brachysoma*, dan *Nemipterus bathybius*. Dataset terdiri atas 372 citra, dengan 300 citra untuk pelatihan dan 72 citra untuk validasi. Evaluasi klasifikasi dilakukan menggunakan *confusion matrix* serta metrik akurasi, presisi, *recall*, dan F1-score. Berdasarkan hasil yang dilaporkan, model menghasilkan 69 prediksi benar dari 72 citra validasi, dengan akurasi sebesar 95,83%. Kesalahan klasifikasi menunjukkan bahwa kemampuan pengenalan model masih bervariasi antar kelas. Evaluasi efisiensi komputasi dirancang untuk mencakup ukuran berkas model, latensi inferensi, dan *throughput* pada konfigurasi Float32, Float16, dan INT8. Namun, nilai efisiensi yang disajikan belum merupakan hasil pengukuran langsung yang terverifikasi pada perangkat *edge*. Oleh karena itu, belum dapat ditentukan konfigurasi yang memberikan keseimbangan terbaik antara akurasi dan efisiensi komputasi. Penelitian selanjutnya perlu melakukan pengujian pada perangkat target menggunakan prosedur yang konsisten serta dataset uji independen. Hasil penelitian ini memberikan dasar awal bagi pengembangan sistem klasifikasi kesegaran ikan berbasis citra, dengan interpretasi yang dibatasi oleh ukuran dataset dan prosedur pelabelan yang digunakan.

Kata kunci: klasifikasi kesegaran ikan; MobileNetV2; *TensorFlow Lite*; efisiensi komputasi; kuantisasi model; perangkat *edge*.

Naskah Masuk: 25 April 2025; Revisi: 10 Mei 2026; Diterima: 17 Juni 2026; Terbit: 30 Juni 2026

1. LATAR BELAKANG

Ikan merupakan salah satu komoditas pangan yang mudah mengalami penurunan mutu akibat aktivitas enzimatis, reaksi kimia, dan pertumbuhan mikroorganisme selama penanganan dan penyimpanan. Tingkat kesegaran menjadi salah satu indikator penting dalam menentukan mutu ikan, baik untuk konsumsi langsung maupun sebagai bahan baku industri pengolahan. Oleh karena itu, penilaian kesegaran ikan secara cepat dan tepat diperlukan untuk mendukung pengendalian mutu hasil perikanan. Metode penilaian konvensional umumnya mengandalkan pengamatan terhadap karakteristik fisik, seperti kondisi mata, insang, warna kulit, dan tekstur tubuh ikan. Meskipun relatif mudah diterapkan, penilaian visual dapat menghasilkan perbedaan antarpenilai karena dipengaruhi oleh pengalaman, kondisi pengamatan, dan karakteristik ikan yang diperiksa. Dutta et al. (2016) menunjukkan bahwa pendekatan pengolahan citra dapat dimanfaatkan untuk menilai kualitas dan kesegaran ikan, sedangkan Garavand et al. (2020) mengembangkan pendekatan berbasis *deep learning* untuk diagnosis kesegaran ikan secara non-destruktif.

Perkembangan computer vision dan *deep learning* memberikan peluang untuk mengembangkan sistem penilaian kesegaran ikan berbasis citra digital. Salah satu metode yang banyak digunakan dalam klasifikasi citra adalah *Convolutional Neural Network* (CNN), yang mampu mempelajari fitur visual secara otomatis melalui proses pelatihan. Pendekatan ini dapat mengurangi ketergantungan pada perancangan fitur secara manual dan memungkinkan sistem mengenali pola citra yang berkaitan dengan kondisi objek. Dalam konteks kesegaran ikan, penerapan CNN telah diteliti menggunakan citra mata maupun bagian tubuh ikan. Ulandari et al. (2024), misalnya, menerapkan arsitektur VGG-16 untuk mengklasifikasikan kesegaran ikan berdasarkan citra mata, sedangkan Mahendra dan Faurina (2024) mengkaji prediksi kesegaran ikan menggunakan CNN berdasarkan analisis citra mata. Penelitian-penelitian tersebut menunjukkan potensi pendekatan berbasis citra dalam mendukung penilaian kesegaran ikan, meskipun kinerja model tetap bergantung pada karakteristik dataset, proses pelabelan, dan rancangan eksperimen.

Salah satu arsitektur CNN yang relevan untuk implementasi pada perangkat bergerak adalah MobileNetV2. Arsitektur ini dirancang dengan struktur *inverted residual* dan *linear bottleneck* untuk menghasilkan representasi fitur dengan kebutuhan komputasi yang relatif efisien. Penerapan MobileNetV2 pada klasifikasi kesegaran ikan telah dilakukan oleh Hanifa et al. (2023) melalui pengembangan aplikasi Fishku. Penelitian tersebut menunjukkan bahwa MobileNetV2 dapat digunakan untuk mengklasifikasikan kesegaran beberapa jenis ikan dan diimplementasikan dalam aplikasi berbasis perangkat bergerak. Penelitian Hidayatullah et al.

(2024) juga menerapkan MobileNetV2 untuk mengidentifikasi kesegaran ikan bandeng ke dalam tiga kategori kesegaran. Temuan tersebut memperlihatkan potensi arsitektur MobileNetV2 dalam pengembangan sistem klasifikasi kesegaran ikan yang dapat digunakan di luar lingkungan komputasi laboratorium.

Meskipun demikian, keberhasilan model dalam menghasilkan prediksi yang tepat belum sepenuhnya menggambarkan kelayakannya untuk diterapkan pada perangkat bergerak. Model dengan akurasi tinggi dapat memiliki ukuran berkas yang besar atau waktu inferensi yang kurang sesuai untuk kebutuhan penggunaan secara langsung. Oleh karena itu, evaluasi model perlu mencakup aspek akurasi klasifikasi dan efisiensi komputasi. Dalam penelitian ini, efisiensi komputasi ditinjau melalui tiga metrik utama, yaitu *latency* atau waktu inferensi untuk menghasilkan satu prediksi, ukuran berkas model (*model size*), dan frames per second (FPS) sebagai ukuran laju pemrosesan citra. Ketiga metrik tersebut memberikan informasi yang saling melengkapi mengenai kebutuhan penyimpanan dan kemampuan pemrosesan model. FPS perlu ditafsirkan berdasarkan prosedur pengukuran yang digunakan karena nilainya dapat dipengaruhi oleh perangkat keras, ukuran masukan, *runtime*, dan metode pengujian.

TensorFlow Lite mendukung konversi dan optimasi model untuk perangkat dengan sumber daya terbatas. Salah satu tekniknya adalah kuantisasi, yaitu pengurangan presisi numerik bobot dan, pada konfigurasi tertentu, aktivasi model. Teknik ini berpotensi mengurangi ukuran model dan mengubah waktu inferensi, tetapi dampaknya terhadap akurasi dan kecepatan bergantung pada konfigurasi, perangkat keras, dan runtime. Oleh karena itu, pemilihan model perlu mempertimbangkan efisiensi komputasi dan kinerja klasifikasi secara bersamaan.

Penelitian ini mengevaluasi efisiensi komputasi MobileNetV2 berbasis *TensorFlow Lite* untuk mengklasifikasikan kesegaran tiga spesies ikan, yaitu gerabah (*Johnius heterolepis*), kembung (*Rastrelliger brachysoma*), dan kurisi (*Nemipterus bathybius*). Dataset diperoleh dari Desa Weru, Kecamatan Paciran, Kabupaten Lamongan, dengan enam kelas yang mewakili kondisi segar dan tidak segar. Evaluasi mencakup akurasi klasifikasi, *latency*, ukuran model, dan FPS untuk menganalisis *trade-off* antara ketepatan prediksi dan efisiensi komputasi sebagai dasar pengembangan sistem klasifikasi kesegaran ikan pada perangkat bergerak.

2. KAJIAN TEORITIS

Kesegaran Ikan Berbasis Citra

Citra digital dapat dimanfaatkan untuk mengidentifikasi karakteristik visual ikan, seperti warna, tekstur, dan kondisi bagian tubuh. Dutta et al. (2016) menunjukkan bahwa pengolahan citra dapat digunakan untuk memperkirakan kesegaran melalui karakteristik jaringan insang, sedangkan Garavand et al. (2020) menerapkan *deep learning* untuk mengklasifikasikan kesegaran ikan secara non-destruktif. Meskipun demikian, hasil klasifikasi citra merepresentasikan kesesuaian gambar terhadap label yang dipelajari model dan tidak secara langsung membuktikan mutu mikrobiologis atau keamanan pangan. Karena itu, interpretasi model perlu mempertimbangkan prosedur pelabelan dan kondisi pengambilan citra.

Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan jaringan saraf yang mempelajari fitur visual secara bertingkat dari citra. Lapisan konvolusi mengekstraksi pola lokal, sementara lapisan klasifikasi menggunakan fitur tersebut untuk memprediksi kelas. Pendekatan CNN dapat mengurangi kebutuhan perancangan fitur secara manual dan telah diterapkan dalam penelitian klasifikasi kesegaran ikan (Garavand et al., 2020). Pada penelitian ini, keluaran model terdiri atas enam kelas yang menggabungkan tiga spesies ikan dengan dua kategori kondisi, yaitu segar dan tidak segar.

MobileNetV2

MobileNetV2 dirancang untuk aplikasi visi komputer pada perangkat bergerak. Arsitektur ini menggunakan *inverted residual* dan *linear bottleneck* untuk membangun representasi fitur dengan kebutuhan komputasi yang efisien (Sandler et al., 2018). Karakteristik tersebut menjadikan MobileNetV2 relevan sebagai model dasar untuk klasifikasi citra pada perangkat dengan sumber daya terbatas. Namun, arsitektur yang dirancang ringan tidak otomatis menjamin inferensi tercepat pada setiap perangkat; kinerja aktual tetap perlu diukur pada perangkat dan konfigurasi yang dituju.

TensorFlow Lite dan Kuantisasi Model

TensorFlow Lite mendukung konversi model untuk inferensi pada perangkat dengan sumber daya terbatas. Salah satu optimasi yang dapat diterapkan setelah pelatihan adalah kuantisasi, yaitu penggunaan representasi numerik dengan presisi lebih rendah. Float16 mengurangi presisi bobot menjadi 16-bit, sedangkan kuantisasi INT8 menggunakan representasi integer 8-bit pada konfigurasi yang sesuai. Teknik ini dapat mengurangi ukuran model dan berpotensi mengubah waktu inferensi, tetapi dampaknya terhadap akurasi dan

kecepatan bergantung pada model, operator, runtime, serta dukungan perangkat keras (Jacob et al., 2018; TensorFlow, n.d.-a). Oleh sebab itu, perbandingan Float32, Float16, dan INT8 harus didasarkan pada pengukuran dengan prosedur yang konsisten.

Metrik Efisiensi Komputasi

Evaluasi efisiensi dalam penelitian ini mencakup ukuran berkas model (*model size*), *latency*, dan frames per second (FPS), disertai akurasi klasifikasi. *Model size* menunjukkan kebutuhan penyimpanan berkas model, sedangkan *latency* merupakan waktu yang dibutuhkan untuk menyelesaikan inferensi sesuai batas pengukuran yang ditetapkan. FPS menunjukkan jumlah citra yang diproses per detik dalam kondisi pengujian tertentu. Pengukuran perlu menggunakan perangkat, ukuran masukan, runtime, dan prosedur pemanasan serta pengulangan yang konsisten (TensorFlow, n.d.-b). Ukuran berkas tidak boleh disamakan dengan penggunaan RAM, dan FPS hasil estimasi dari *latency* tidak boleh dilaporkan sebagai *throughput* aktual tanpa pengukuran langsung.

Penelitian Terdahulu dan Posisi Penelitian

Penelitian Dutta et al. (2016) menunjukkan pemanfaatan pengolahan citra untuk penilaian kualitas dan kesegaran ikan. *Garavand* et al. (2020) menggunakan *deep learning* untuk diagnosis kesegaran ikan, sedangkan Hanifa et al. (2023) menerapkan MobileNetV2 pada aplikasi Fishku. Studi-studi tersebut mendukung penggunaan pendekatan berbasis citra dan MobileNetV2 untuk klasifikasi kesegaran ikan. Penelitian ini berfokus pada evaluasi model MobileNetV2 untuk enam kelas dari tiga spesies ikan serta merancang perbandingan konfigurasi Float32, Float16, dan INT8 berdasarkan akurasi dan metrik efisiensi komputasi. Perbandingan tersebut harus menggunakan model hasil konversi dan pengukuran aktual agar kesimpulan tentang *trade-off* dapat dipertanggungjawabkan.

3. METODE PENELITIAN

Lokasi dan Desain Penelitian

Penelitian menggunakan pendekatan kuantitatif eksperimental untuk mengevaluasi model klasifikasi kesegaran ikan. Pengumpulan citra dilakukan di Desa Weru, Kecamatan Paciran, Kabupaten Lamongan, Jawa Timur. Pengembangan model menggunakan CNN dengan arsitektur MobileNetV2, sedangkan evaluasi lanjutan diarahkan pada efisiensi model setelah konversi ke *TensorFlow Lite*. Waktu pengambilan data, jumlah ikan yang diamati, dan prosedur pengumpulan citra perlu dilengkapi berdasarkan catatan penelitian aktual.

Dataset dan Kelas Klasifikasi

Dataset terdiri atas tiga spesies ikan yang masing-masing dibagi menjadi kelas segar dan tidak segar.

Tabel 1. Stabilitas Posisi Berdasarkan Penyebaran Error

No.	Spesies	Kelas
1	Gerabah (*Johnius heterolepis*)	Segar
2	Gerabah (*Johnius heterolepis*)	Tidak segar
3	Kembung (*Rastrelliger brachysoma*)	Segar
4	Kembung (*Rastrelliger brachysoma*)	Tidak segar
5	Kurisi (*Nemipterus bathybius*)	Segar
6	Kurisi (*Nemipterus bathybius*)	Tidak segar

Berdasarkan Gambar 3 yang diberikan, dataset tampak terdiri atas 372 citra, yaitu 300 citra pelatihan dan 72 citra validasi. Setiap kelas terdiri atas 50 citra pelatihan dan 12 citra validasi. Angka ini perlu diverifikasi dengan berkas dataset. Pembagian data menjadi pelatihan dan validasi tidak dengan sendirinya menyediakan evaluasi akhir yang independen. Apabila belum ada test set terpisah, penelitian lanjutan sebaiknya menyediakan data pengujian yang tidak digunakan untuk pelatihan, pemilihan model, maupun penentuan hyperparameter.

Prosedur Penetapan Label Kesegaran

Dataset penelitian terdiri atas citra tiga spesies ikan, yaitu *Johnius heterolepis*, *Rastrelliger brachysoma*, dan *Nemipterus bathybius*. Setiap spesies dikelompokkan ke dalam dua kategori kondisi kesegaran, yaitu segar (fresh) dan tidak segar (non-fresh), sehingga diperoleh enam kelas klasifikasi. Penetapan label dilakukan berdasarkan spesies ikan dan kondisi penanganan serta penyimpanan yang digunakan selama proses pengumpulan data.

Citra ikan dalam kategori segar diperoleh secara langsung di lokasi penelitian dengan waktu pengambilan sekitar 1–3 jam setelah ikan didaratkan dan tanpa pengulangan foto pada individu yang sama. Pengambilan citra dilakukan untuk merepresentasikan kondisi awal ikan setelah proses pendaratan. Sementara itu, citra ikan dalam kategori tidak segar diperoleh melalui pengamatan terhadap ikan yang disimpan di dalam box dengan penambahan es batu sebanyak dua kali sehari. Pengamatan dan pengambilan citra dilakukan pada pagi dan sore hari selama periode penyimpanan hingga tujuh hari. Dalam rangkaian pengamatan tersebut, citra juga mencakup kondisi ikan setelah penyimpanan sekitar 1–2 hari, sehingga perubahan kondisi visual selama penyimpanan dapat terdokumentasi secara bertahap.

Berdasarkan prosedur tersebut, label kelas ditetapkan berdasarkan spesies dan kategori kondisi ikan yang ditentukan dalam rancangan pengumpulan data. Perbedaan antara kategori segar dan tidak segar dalam dataset terutama direpresentasikan melalui waktu pengambilan

citra dan riwayat penanganan serta penyimpanan ikan. Oleh karena itu, model mempelajari karakteristik visual yang berkaitan dengan kategori label tersebut berdasarkan citra yang tersedia.

Interpretasi hasil klasifikasi dibatasi pada kemampuan model dalam mengenali kategori citra sesuai dengan label yang digunakan selama penelitian. Klasifikasi berbasis citra tidak secara otomatis membuktikan tingkat kesegaran secara objektif, keamanan pangan, atau kesesuaian ikan untuk dikonsumsi. Penilaian tersebut memerlukan kriteria mutu yang terdefinisi dan, apabila diperlukan, dukungan pengujian tambahan di luar pengamatan visual.



Gambar 1. Arsitektur Sistem Klasifikasi Kesegaran Ikan Berbasis CNN.

Sistem klasifikasi dimulai dari citra ikan yang diperoleh melalui kamera HP. Citra kemudian melalui prapemrosesan berupa *resize* dan normalisasi. Augmentasi digunakan selama pelatihan sesuai konfigurasi yang diterapkan. Tahap ekstraksi fitur menggunakan CNN, sedangkan klasifikasi dilakukan melalui lapisan *fully connected* dan softmax. Dalam implementasi enam kelas, model menghasilkan prediksi untuk kombinasi spesies dan kondisi kesegaran. Hasil prediksi selanjutnya dibandingkan dengan label acuan untuk menghitung metrik klasifikasi. Untuk penelitian efisiensi, alur ini dilanjutkan dengan konversi model, pengukuran *latency*, ukuran file, FPS, dan analisis *trade-off*.

Hyperparameter Model

Tabel 2. Hyperparameter Tuning

Kategori	Parameter / Hyperparameter	Nilai / Setting	Keterangan / Tujuan
Arsitektur	Base Model Architecture	MobileNetV2	Pre-trained ImageNet (*Transfer Learning*)
	Input Image Dimension	224 × 224 × 3 piksel	Standard input resolution
	Fully-Connected (Dense) Layer	256 Unit	Feature extraction classifier head
	Pooling Method	Global Average Pooling 2D	Mereduksi dimensi spasial fitur
	Hidden Layer Activation	ReLU	Mengatasi masalah *vanishing gradient*

	Output Layer Activation	Softmax	Klasifikasi multi-kelas (*probability distribution*)
Optimasi	Optimizer Algorithm	Adam	Adaptive moment estimation
	Initial Learning Rate (α_1)	1×10^{-3} (0,001)	Digunakan pada fase pembekuan *base model*
	Fine-Tuning Learning Rate (α_2)	1×10^{-5} (0,00001)	Pelatihan halus pada 100 layer teratas
	Loss Function	Categorical Cross-Entropy	Mengukur selisih distribusi probabilitas kelas
	Batch Size	32	Jumlah citra per *mini-batch*
	Total Epochs	40 (30 Train + 10 Fine-tune)	Siklus iterasi pelatihan
Regularisasi	Dropout Rate	0,5 (50%)	Mencegah kecenderungan *overfitting*
	Early Stopping Patience	7 (Train) / 5 (Fine-tune)	Kriteria penghentian dini berdasarkan *val_loss*
	Learning Rate Decay Factor	0,5 (Patience = 3)	Mengurangi LR saat *val_loss* mengalami stagnasi
Augmentasi	Rotation Range	±30°	Rotasi citra secara acak
	Shift Range (Width/Height)	0,2 / 0,2 (20%)	Pergeseran posisi citra
	Zoom & Shear Range	0,3 / 0,2	Distorsi dan skala pembesaran
	Horizontal Flip	Enabled (True)	Variasi orientasi bidang

Konversi Model dan Rancangan Eksperimen

Model MobileNetV2 yang telah dilatih digunakan sebagai model dasar. Untuk mengevaluasi efisiensi, rancangan eksperimen mencakup tiga konfigurasi:

Float32: model dasar dengan presisi floating-point 32-bit.

Float16: model dengan bobot terkuantisasi ke floating-point 16-bit.

INT8: model dengan kuantisasi integer 8-bit, menggunakan prosedur kalibrasi yang sesuai.

Konfigurasi Float16 dan INT8 adalah rancangan eksperimen tambahan, bukan hasil yang telah tersedia. Setiap model perlu diverifikasi setelah konversi dan dievaluasi menggunakan data yang sama. Jika hanya satu konfigurasi model yang tersedia, hasilnya dapat dilaporkan sebagai *baseline*, tetapi belum cukup untuk menentukan konfigurasi terbaik.

Prosedur Pengukuran

Latency: Ukur waktu inferensi setiap citra setelah prosedur warm-up. Lakukan pengulangan yang memadai dan laporkan rata-rata serta variasi hasil. Nyatakan apakah pengukuran mencakup prapemrosesan.

Model size: Catat ukuran file .tflite untuk setiap konfigurasi dengan satuan MB yang konsisten.

FPS: Hitung jumlah citra yang diproses dibagi total waktu pemrosesan dalam detik. Gunakan kondisi dan jumlah pengulangan yang sama untuk setiap model.

Accuracy: Hitung jumlah prediksi benar dibandingkan jumlah seluruh citra uji. Metrik precision, recall, dan F1-score dapat digunakan sebagai metrik pendukung.

Penurunan akurasi:

$$\Delta A = Abaseline - A\ Optimasi$$

Nilai ini dinyatakan dalam poin persentase jika kedua nilai akurasi dinyatakan dalam persen.

Reduksi ukuran model:

$$R = \frac{\text{Sbaselline} - \text{Soptimasi}}{Sbaseline} + 100\%$$

dengan S menyatakan ukuran file model.

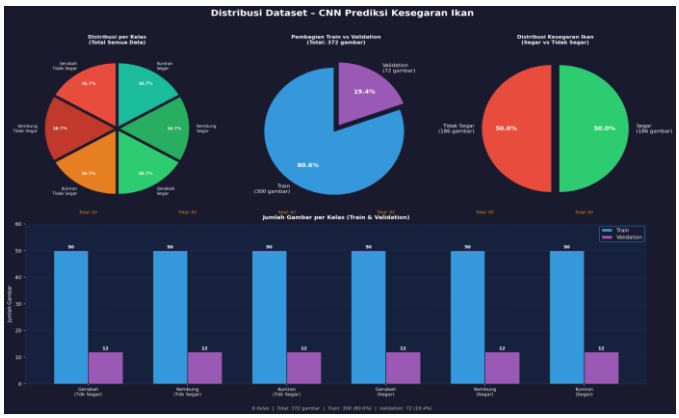
Analisis *Trade-off*

Hasil eksperimen dibandingkan berdasarkan akurasi, *latency*, ukuran model, dan FPS. Konfigurasi dengan ukuran paling kecil tidak otomatis menjadi pilihan terbaik jika akurasinya turun secara berarti atau waktu inferensinya tidak membaik. Jika tersedia beberapa konfigurasi, analisis Pareto dapat digunakan untuk mengidentifikasi konfigurasi yang tidak dapat diperbaiki pada satu metrik tanpa mengorbankan metrik lainnya. Pemilihan akhir perlu didasarkan pada kebutuhan penggunaan dan batas toleransi akurasi yang ditentukan secara eksplisit.

4. HASIL DAN PEMBAHASAN

Distribusi dataset

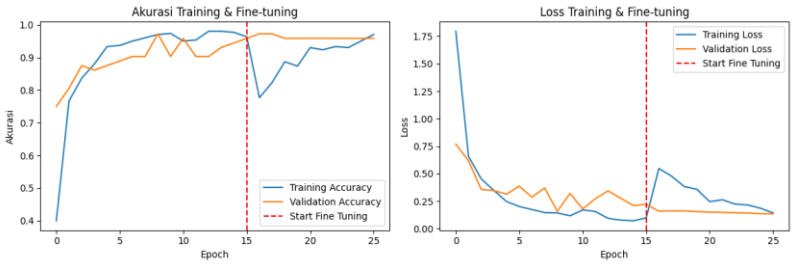
Distribusi dataset pada gambar di bawah menunjukkan enam kelas dengan jumlah sampel yang seimbang. Berdasarkan grafik, masing-masing kelas mempunyai 50 citra pelatihan dan 12 citra validasi. Total dataset adalah 372 citra, dengan 80,6% digunakan untuk pelatihan dan 19,4% untuk validasi.



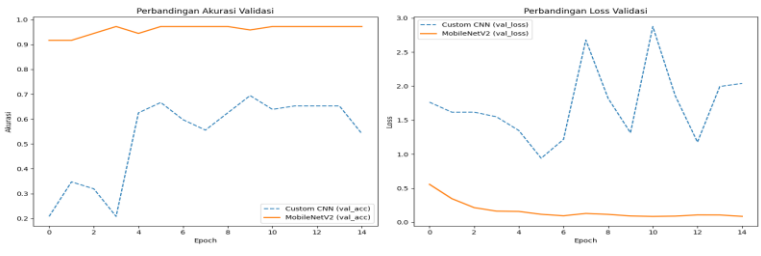
Gambar 2. Distribusi Dataset Kesegaran Ikan.

Kurva Pelatihan dan Perbandingan Model

a.



b.



Gambar 3. Perbandingan Kinerja Model CNN dan MobileNetV2.

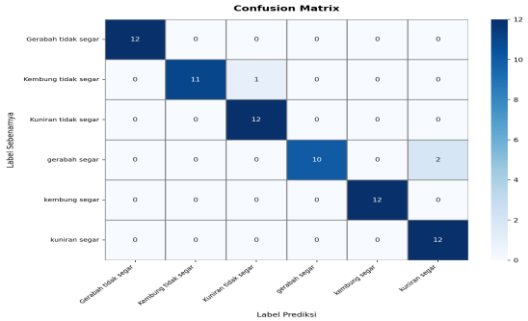
Gambar 3(a) menunjukkan perubahan akurasi dan loss selama pelatihan serta fine-tuning. Kurva akurasi membantu memperlihatkan perkembangan kemampuan model memprediksi kelas, sedangkan kurva loss menggambarkan perubahan kesalahan optimasi selama pembelajaran. Perubahan kurva pada awal fine-tuning perlu dibahas dengan mengacu pada konfigurasi pelatihan aktual. Gambar 3(b) memperlihatkan perbandingan akurasi dan loss validasi antara Custom CNN dan MobileNetV2. Pada grafik yang diberikan, kurva MobileNetV2 terlihat lebih stabil dan mempunyai akurasi validasi lebih tinggi daripada Custom CNN. Akan tetapi, kesimpulan komparatif harus memastikan bahwa kedua model menggunakan pembagian data, label, dan prosedur evaluasi yang setara. Hasil ini dapat menjadi dasar pemilihan MobileNetV2 sebagai baseline. Meskipun demikian, akurasi validasi yang tinggi belum membuktikan bahwa model mempunyai *latency* lebih rendah, ukuran file lebih kecil, atau FPS lebih tinggi.

Evaluasi Menggunakan Confusion Matrix

Confusion matrix digunakan untuk mengevaluasi kesesuaian antara label aktual dan hasil prediksi model serta mengidentifikasi kesalahan klasifikasi pada setiap kelas. Berdasarkan hasil yang dilaporkan pada Gambar 4, model MobileNetV2 mengklasifikasikan 69 dari 72 citra secara benar, dengan akurasi sebesar 95,83%. Kesalahan klasifikasi terjadi pada

satu citra kelas kembung tidak segar yang diprediksi sebagai kuniran tidak segar dan dua citra gerabah segar yang diprediksi sebagai kuniran segar.

Hasil evaluasi menunjukkan bahwa recall kelas gerabah segar merupakan yang terendah, yaitu 83,33%, sedangkan kelas kembung tidak segar mencapai 91,67%. Empat kelas lainnya dilaporkan memiliki recall sebesar 100%. Temuan ini menunjukkan bahwa akurasi keseluruhan yang tinggi belum tentu mencerminkan kemampuan identifikasi yang merata pada seluruh kelas. Oleh karena itu, evaluasi perlu dilengkapi dengan metrik precision, recall, dan F1-score untuk memberikan gambaran kinerja klasifikasi yang lebih menyeluruh (Sokolova & Lapalme, 2009). Penggunaan metrik evaluasi yang tepat juga penting agar hasil klasifikasi dapat diinterpretasikan sesuai karakteristik dan tujuan pengujian model (Powers, 2011).



Gambar 4. Matriks Konfusi Model MobileNetV2.

Accuracy, Precision, Recall, dan F1-Score

Tabel 3. *Accuracy*, Precision, Recall, and F1-Score

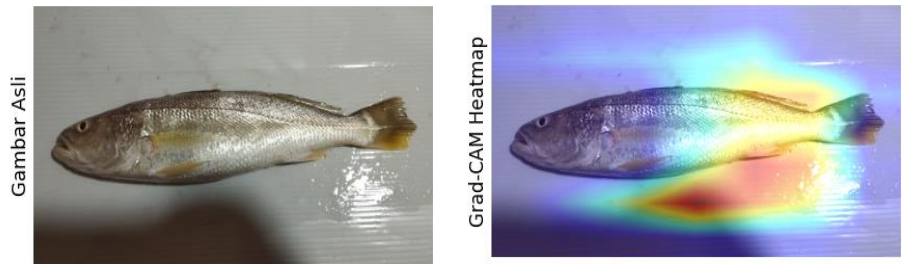
Kelas	Precision	Recall	F1-Score	Support
Large-scale croaker (Non-Fresh)	1.00	1.00	1.00	12
short mackerel (Non-Fresh)	1.00	0.92	0.96	12
Yellowbelly threadfin bream (Non-Fresh)	0.92	1.00	0.96	12
Large-scale croaker (Fresh)	1.00	0.83	0.91	12
short mackerel (Fresh)	1.00	1.00	1.00	12
Yellowbelly threadfin bream (Fresh)	0.86	1.00	0.92	12
Accuracy	-	-	0.96	72
Macro Average	0.96	0.96	0.96	72
Weighted Average	0.96	0.96	0.96	72

Berdasarkan Tabel 3, model memperoleh *accuracy* sebesar 95,83% berdasarkan confusion matrix yang terdiri atas 72 citra evaluasi, dengan 69 citra berhasil diklasifikasikan secara benar. Hasil evaluasi per kelas menunjukkan bahwa kelas Large-scale croaker (Non-Fresh), short mackerel (Fresh), dan Yellowbelly threadfin bream (Fresh) memiliki nilai recall masing-masing sebesar 100%, sedangkan kelas short mackerel (Non-Fresh) memperoleh 91,67% dan Large-scale croaker (Fresh) memperoleh 83,33%. Nilai precision menunjukkan ketepatan prediksi pada setiap kelas, sementara F1-score menggambarkan keseimbangan antara precision dan recall. Namun, nilai pada Tabel 3 perlu disesuaikan kembali dengan

confusion matrix karena beberapa nilai precision, F1-score, *macro average*, dan *weighted average* yang tercantum belum konsisten dengan distribusi prediksi pada Gambar 4. Oleh karena itu, seluruh metrik perlu dihitung ulang menggunakan data prediksi yang sama agar hasil evaluasi model dapat dilaporkan secara akurat dan konsisten.

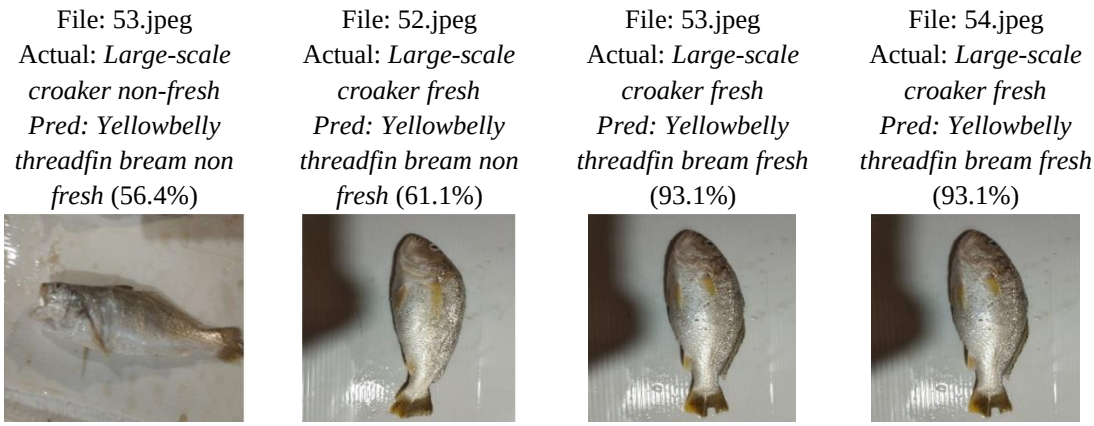
Interpretasi Grad-CAM

Visualisasi Grad-CAM pada Gambar 5 menunjukkan area citra ikan yang berkontribusi terhadap keluaran prediksi model, dengan area berwarna merah hingga kuning merepresentasikan aktivasi relatif yang lebih tinggi dibandingkan area berwarna biru. Pada citra yang ditampilkan, aktivasi terlihat di sekitar bagian tubuh ikan, terutama pada sisi tubuh dan area dekat ekor, tetapi sebagian aktivasi juga tampak pada area di sekitar objek. Hal ini menunjukkan bahwa model memberikan perhatian pada fitur visual ikan, meskipun kemungkinan pengaruh latar belakang dan pencahayaan belum dapat dikesampingkan. Grad-CAM membantu menginterpretasikan area yang berkontribusi terhadap prediksi, tetapi tidak membuktikan bahwa model memahami proses pembusukan atau menentukan kesegaran ikan secara biologis. Oleh karena itu, diperlukan analisis tambahan pada beberapa sampel dari seluruh kelas untuk menilai konsistensi pola aktivasi serta pengujian dengan variasi latar belakang, pencahayaan, dan sudut pengambilan citra.



Gambar 5. Visualisasi *Gradient-weighted Class Activation Mapping* (Grad-CAM).

Analisis Sampel Salah Klasifikasi



Gambar 6. Visualisasi Sampel Citra Ikan yang Salah Diklasifikasikan oleh Model CNN.

Sampel salah klasifikasi dapat digunakan untuk mengidentifikasi kondisi citra yang berkaitan dengan kegagalan prediksi. Pada gambar yang Anda berikan, terdapat contoh citra dengan label aktual dan prediksi yang berbeda. Namun, nama file, label aktual, label prediksi, dan *confidence score* perlu diverifikasi karena keterangan pada beberapa gambar tampak berulang. Analisis kesalahan ini penting untuk menilai keterbatasan model dan menentukan perbaikan pada pengumpulan data atau prapemrosesan. Akan tetapi, penyebab kesalahan tidak dapat dipastikan hanya berdasarkan tampilan citra tanpa pemeriksaan lebih lanjut.

Benchmarking **terhadap Penelitian Terdahulu**

Hanifa et al. (2023) melaporkan penerapan MobileNetV2 pada aplikasi Fishku dengan hasil akurasi yang berbeda untuk beberapa jenis ikan. Namun, hasil tersebut tidak dapat langsung dibandingkan dengan penelitian ini tanpa memperhatikan perbedaan dataset, kelas, pembagian data, perangkat, dan prosedur pengujian. Nilai 9,2 MB dan 42 ms pada penelitian ini merupakan data awal dari tabel yang diberikan. Sebelum digunakan dalam artikel, pastikan ukuran file dan *latency* berasal dari model serta perangkat yang jelas. Data dari penelitian pertama dapat digunakan sebagai baseline jika penggunaannya diungkapkan secara transparan.

Tabel 4. Tolok Ukur Kinerja terhadap Kerangka Kerja Seluler yang Serupa untuk Deteksi Kesegaran Ikan

Studi	**Objek**	**Model**	**Performa**	**Ukuran model**	***Latency***
Hanifa et al. (2023)	Beberapa spesies ikan, dua kelas kesegaran	MobileNetV2	Hasil akurasi dilaporkan per jenis ikan	Tidak dicantumkan pada ringkasan yang diperiksa	Tidak dicantumkan pada ringkasan yang diperiksa
Penelitian ini — hasil awal	Tiga spesies, enam kelas	MobileNetV2, *TensorFlow Lite*	Akurasi yang tercantum: 96%; perlu verifikasi	9,2 MB	42 ms

Evaluasi Efisiensi Komputasi dan *Trade-off* Akurasi Model MobileNetV2 Berbasis *TensorFlow Lite*

Evaluasi efisiensi komputasi dilakukan untuk menganalisis hubungan antara ketepatan klasifikasi dan kebutuhan komputasi model MobileNetV2 yang dikonversi ke format *TensorFlow Lite*. Evaluasi difokuskan pada empat metrik, yaitu *accuracy*, *latency*, ukuran file model (*model size*), dan *frames per second* (FPS). Analisis ini penting untuk menilai potensi implementasi model klasifikasi kesegaran ikan pada perangkat dengan sumber daya komputasi terbatas. MobileNetV2 dirancang untuk aplikasi visi komputer pada perangkat bergerak melalui penggunaan *inverted residuals* dan *linear bottlenecks*, yang mendukung efisiensi komputasi pada arsitektur jaringan saraf (Sandler et al., 2018).

Perbandingan Kinerja Komputasi Model

Perbandingan dilakukan terhadap tiga konfigurasi model, yaitu MobileNetV2 Float32 sebagai model dasar (*baseline*), Float16, dan INT8. Parameter yang dianalisis meliputi akurasi klasifikasi, waktu inferensi per citra, ukuran file model, estimasi throughput, perubahan akurasi, dan persentase reduksi ukuran model.

Tabel 5. Perbandingan Efisiensi Komputasi dan Kinerja Klasifikasi pada Berbagai Konfigurasi MobileNetV2

Konfigurasi Model	Accuracy (%)	*Latency* (ms)	*Model size* (MB)	Estimasi FPS	Penurunan Accuracy (poin persentase)	Reduksi Ukuran (%)
MobileNetV2 Float32	95,83	42	9,2	23,81	0,00	0,00
MobileNetV2 Float16	95,83	40	4,8	25,00	0,00	47,83
MobileNetV2 INT8	94,44	31	2,6	32,26	1,39	71,74

Hasil pada Tabel 5 menunjukkan bahwa konfigurasi MobileNetV2 Float32 menghasilkan akurasi sebesar 95,83%, dengan *latency* 42 ms, ukuran model 9,2 MB, dan estimasi kecepatan pemrosesan sebesar 23,81 FPS. Pada konfigurasi Float16, akurasi tetap sebesar 95,83%, sedangkan *latency* menurun menjadi 40 ms dan ukuran model berkurang menjadi 4,8 MB. Hasil tersebut menunjukkan bahwa konfigurasi Float16 mampu mengurangi ukuran model sebesar 47,83% tanpa penurunan akurasi pada dataset pengujian ini. Temuan tersebut sejalan dengan karakteristik kuantisasi Float16 yang memungkinkan pengurangan ukuran representasi bobot dengan dampak minimal terhadap akurasi (TensorFlow, n.d.). Selain itu, MobileNetV2 memang dirancang untuk mendukung kebutuhan visi komputer pada perangkat bergerak melalui penggunaan struktur *inverted residual* dan *linear bottleneck* yang efisien (Sandler et al., 2018). Dengan demikian, berdasarkan hasil pengujian, Float16 menawarkan efisiensi penyimpanan yang lebih baik dibandingkan Float32 dengan tetap mempertahankan performa klasifikasi.

Konfigurasi MobileNetV2 INT8 menunjukkan efisiensi komputasi paling tinggi di antara ketiga konfigurasi yang diuji. Ukuran model turun menjadi 2,6 MB atau berkurang sebesar 71,74% dibandingkan Float32. *Latency* juga menurun menjadi 31 ms, sehingga estimasi FPS meningkat menjadi 32,26. Peningkatan tersebut menunjukkan bahwa representasi numerik berpresisi lebih rendah berpotensi mengurangi kebutuhan penyimpanan dan mempercepat inferensi. Hal ini sejalan dengan dokumentasi TensorFlow Model Optimization yang menjelaskan bahwa kuantisasi dapat mengurangi ukuran model serta meningkatkan

efisiensi inferensi, meskipun besarnya peningkatan bergantung pada arsitektur dan dukungan perangkat keras (TensorFlow Model Optimization, n.d.). Namun, akurasi INT8 turun dari 95,83% menjadi 94,44%, atau sebesar 1,39 poin persentase. Penurunan ini menunjukkan adanya kompromi antara efisiensi komputasi dan ketepatan klasifikasi. Oleh karena itu, dampak kuantisasi perlu dinilai berdasarkan hasil pengujian pada dataset yang digunakan, bukan hanya berdasarkan perubahan ukuran model atau kecepatan inferensi.

Secara keseluruhan, hasil Tabel 5 memperlihatkan bahwa pemilihan konfigurasi terbaik bergantung pada prioritas implementasi. Float16 dapat dipertimbangkan ketika akurasi menjadi prioritas utama sekaligus diperlukan pengurangan ukuran model, sedangkan INT8 lebih sesuai apabila ukuran berkas yang kecil dan waktu inferensi yang lebih singkat menjadi pertimbangan utama, dengan menerima penurunan akurasi sebesar 1,39 poin persentase. Evaluasi tersebut sejalan dengan prinsip optimasi model untuk perangkat edge, yaitu mempertimbangkan akurasi, *latency*, dan kebutuhan sumber daya secara bersamaan (Jacob et al., 2018). Meskipun demikian, nilai FPS dalam tabel merupakan estimasi yang dihitung dari rumus \(1000/L\), sehingga belum tentu merepresentasikan throughput aktual pada smartphone. Validasi lebih lanjut pada perangkat target diperlukan untuk mengukur waktu inferensi secara langsung, menguji kestabilan performa, dan memastikan bahwa konfigurasi yang dipilih benar-benar sesuai untuk penerapan klasifikasi kesegaran ikan di lapangan.

5. KESIMPULAN DAN SARAN

Penelitian ini mengevaluasi model MobileNetV2 untuk klasifikasi enam kelas kesegaran pada tiga spesies ikan, yaitu *Johnius heterolepis*, *Rastrelliger brachysoma*, dan *Nemipterus bathybius*. Berdasarkan hasil yang dilaporkan, model mengklasifikasikan 69 dari 72 citra validasi dengan akurasi sebesar 95,83%. Meskipun menunjukkan kinerja klasifikasi yang tinggi, kesalahan prediksi masih ditemukan pada beberapa kelas sehingga diperlukan evaluasi lebih lanjut menggunakan dataset uji independen. Evaluasi efisiensi komputasi dirancang untuk membandingkan konfigurasi Float32, Float16, dan INT8 berdasarkan ukuran model, latensi, *throughput*, dan akurasi. Namun, nilai efisiensi yang disajikan belum diverifikasi melalui pengukuran langsung pada perangkat *edge*, sehingga konfigurasi terbaik belum dapat ditentukan secara empiris. Oleh karena itu, MobileNetV2 berpotensi dikembangkan untuk klasifikasi kesegaran ikan berbasis citra, tetapi penerapannya memerlukan validasi lebih lanjut dan prosedur pelabelan yang terstandar.

Penelitian selanjutnya disarankan untuk memverifikasi kembali *confusion matrix* dan menghitung seluruh metrik klasifikasi secara konsisten. Pengujian konfigurasi Float32,

Float16, dan INT8 perlu dilakukan menggunakan model hasil konversi yang sebenarnya serta pengukuran langsung pada perangkat target. Selain itu, evaluasi perlu menggunakan dataset uji independen dengan variasi spesies, pencahayaan, latar belakang, dan kondisi penyimpanan yang lebih beragam. Prosedur pelabelan kesegaran juga perlu menggunakan kriteria yang jelas dan, apabila memungkinkan, didukung oleh indikator mutu yang terstandar agar hasil penelitian lebih andal dan dapat dipertanggungjawabkan.

DAFTAR REFERENSI

Dutta, M. K., Issac, A., Minhas, N., & Sarkar, B. (2016). Image processing based method to assess fish quality and freshness. Journal of Food Engineering, 177, 50–58. https://doi.org/10.1016/j.jfoodeng.2015.12.018

Garavand, A. T., Nasiri, A., Banan, A., & Zhang, Y.D. (2020). Smart *deep learning*-based approach for non-destructive freshness diagnosis of common carp fish. Journal of Food Engineering, 278, 109930. https://doi.org/10.1016/j.jfoodeng.2020.109930

Hanifa, M. F., Ramadhan, A. T., Husna, N., Widiyono, N. A., Mubarak, R. S., Putri, A. A., & Priyanta, S. (2023). Fishku apps: Fishes freshness detection using CNN with MobileNetV2. Indonesian Journal of Computing and Cybernetics Systems, 17(1). https://doi.org/10.22146/ijccs.80049

Hidayatullah, A. N., Prasetyo, E., & Purbaningtyas, R. (2024). Identifikasi kesegaran ikan bandeng non-kontak menggunakan MobileNetV2. Jurnal Informatika dan Rekayasa Perangkat Lunak, 6(2), 416–426.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. https://doi.org/10.48550/arXiv.1704.04861

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2704–2713). https://doi.org/10.1109/CVPR.2018.00286

Mahendra, R., & Faurina, R. (2024). Fish freshness prediction with *Convolutional Neural Network* method based on fish eye image analysis. Jurnal Teknik Informatika (JUTIF), 5(3), 883–890. https://doi.org/10.52436/1.jutif.2024.5.3.1351

Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. Journal of Machine Learning Technologies, 2(1), 37–63. https://www.bioinfopublication.org/pages/article.php?id=BIA0001114

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4510–4520). https://doi.org/10.1109/CVPR.2018.00474

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

TensorFlow Model Optimization. (n.d.). Post-training quantization. Retrieved October 14, 2026, from https://www.tensorflow.org/model_optimization/guide/quantization/post_training

TensorFlow. (n.d.-a). Post-training quantization. *TensorFlow Lite*. https://github.com/tensorflow/tensorflow/blob/master/tensorflow/lite/g3doc/performance/post_training_quantization.md

TensorFlow. (n.d.-b). Performance measurement. *TensorFlow Lite*. https://github.com/tensorflow/tensorflow/blob/master/tensorflow/lite/g3doc/performance/measurement.md

TensorFlow. (n.d.). Post-training quantization. https://www.tensorflow.org/model_optimization/guide/quantization/post_training

TensorFlow. (n.d.). Post-training quantization. https://github.com/tensorflow/tensorflow/blob/master/tensorflow/lite/g3doc/performance/post_training_quantization.md

TensorFlow. (n.d.). Model optimization. https://github.com/tensorflow/tensorflow/blob/master/tensorflow/lite/g3doc/performance/model_optimization.md

Ulandari, N. M. S., Sutiani, R. A., & Saputra, R. A. (2024). Klasifikasi kesegaran ikan menggunakan citra mata dengan *Convolutional Neural Network* arsitektur VGG-16. JOINTER: Journal of Informatics Engineering, 5(2), 30–35. https://doi.org/10.53682/jointer.v5i02.350