

Evaluation of YOLOv8 and Faster R-CNN for Image-Based Food Detection

Julian Kiyosaki Hananta¹, Nuri Cahyono²

* Informatika, Universitas Amikom Yogyakarta

kiyosaki10@students.amikom.ac.id¹, nuricahyono@amikom.ac.id²

Article Info

Article history:

Received 2025-11-04

Revised 2025-12-05

Accepted 2025-12-22

Keyword:

Food Detection,

Object Detection,

YOLOv8,

Faster R-CNN,

Convolutional Neural Network.

ABSTRACT

Difficulties in manually tracking nutrition lead to the need for automatic food detection systems. However, Indonesian food presents tough challenges to recognize because similar-looking foods and different serving styles make it hard. This study looks at two deep learning models that follow different approaches: YOLOv8, which is known for being fast and efficient, and Faster R-CNN, which is known for being very accurate. The goal is to find the best model for use on mobile devices. This research uses a strict and standardized way to test the models to make sure the comparison is fair. A public dataset with 1,325 images from Roboflow was used. To deal with uneven class distribution, the images were split using Stratified Random Sampling. Before training, the images were resized using letterbox method to keep their original shape and normalized for pixel values. Both models were trained for the same number of epochs (100) and used the same optimizer (SGD) to ensure fair comparisons. The results show that YOLOv8 performs better in all areas. It achieved 88.6% mAP@50 accuracy and 62.0% mAP@50-95 precision. Faster R-CNN got 85.5% and 55.6% respectively. YOLOv8 also excels in sensitivity or Recall, reaching 87.7% compared to 61.7% for Faster R-CNN. The F1-Score, which balances accuracy and sensitivity, is 84.0% for YOLOv8 and 72% for Faster R-CNN. In terms of speed and size, YOLOv8 is much better. It runs in 13.5 ms and is 21.5 MB in size. That makes it 7.7 times faster and 7.3 times smaller than Faster R-CNN. Based on these results, YOLOv8 is the best choice for developing mobile-based nutrition tracking systems.



This is an open access article under the [CC-BY-SA](#) license.

I. PENDAHULUAN

Masalah gizi dan pola makan tidak sehat telah menjadi tantangan kesehatan global yang mendesak di era modern, dengan implikasi yang meluas dari tingkat individu hingga sistem kesehatan nasional. Peningkatan signifikan prevalensi obesitas, malnutrisi, serta penyakit tidak menular kronis seperti diabetes, penyakit jantung, dan hipertensi sangat erat kaitannya dengan gaya hidup dan pola konsumsi yang kurang seimbang. Penanganan masalah gizi sangat penting karena skrining dan penilaian malnutrisi yang efektif, terutama untuk pasien rawat inap, lansia, dan penderita penyakit kronis, sangat penting untuk mendeteksi risiko gizi buruk sejak dini. Penanganan awal terbukti bisa mengurangi angka penyakit, kematian, durasi rawat inap, serta biaya pengobatan secara

nyata. Fenomena ini bukan hanya sekumpulan statistik, melainkan cerminan dari kegagalan sistemik dalam manajemen diet secara mandiri dan menegaskan bahwa pola makan yang tidak terkontrol, didorong oleh lingkungan pangan modern, menjadi salah satu penyebab utama krisis kesehatan masyarakat modern [1].

Salah satu hambatan utama yang dihadapi masyarakat dalam mengelola pola makan adalah kesulitan dalam proses pemantauan asupan gizi. Metode manual, seperti menghitung kalori menggunakan tabel referensi, terbukti tidak efektif dan tidak berkelanjutan bagi sebagian besar orang. Pendekatan ini bukan hanya melelahkan dan memakan waktu, tetapi juga sangat rentan terhadap ketidakakuratan yang signifikan, baik karena kesalahan dalam estimasi ukuran porsi maupun kesulitan dalam mengidentifikasi komposisi hidangan yang

kompleks. Menjawab tantangan fundamental ini, kemajuan teknologi, terutama dalam bidang computer vision, menawarkan solusi otomatis. Dengan memanfaatkan penetrasi ponsel pintar yang masif, sistem yang mampu mengenali dan mengklasifikasikan jenis makanan secara *real-time* hanya dengan mengarahkan kamera dapat secara drastis menyederhanakan dan mengotomatisasi proses pemantauan gizi. Teknologi ini berpotensi mengubah tugas yang membosankan menjadi interaksi yang instan dan menarik, sehingga lebih mudah diakses dan diadopsi oleh pengguna secara luas [2].

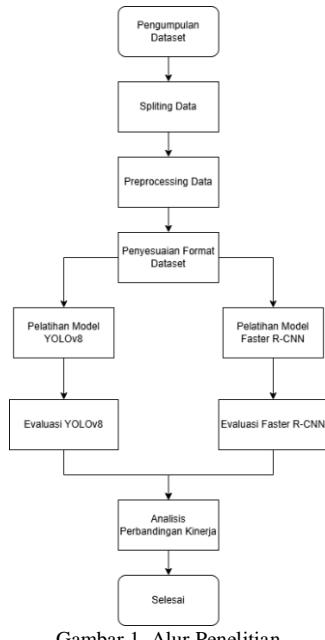
Dalam ranah deteksi makanan berbasis citra, pendekatan *deep learning* telah menjadi standar utama, dengan dua paradigma arsitektur yang dominan, detektor dua tahap (*two-stage*) dan detektor satu tahap (*single-stage*). Arsitektur dua tahap seperti Faster R-CNN dikenal dengan presisinya yang tinggi, berkat pendekatannya yang metodis dalam mengidentifikasi wilayah kandidat objek sebelum melakukan klasifikasi [3]. Arsitektur satu tahap yang diperkenalkan oleh keluarga YOLO (*You Only Look Once*) memberikan kecepatan dalam proses inferensi yang sangat baik, karena mampu menganalisis citra secara keseluruhan sekaligus [4]. Perbandingan antara kedua pendekatan ini telah menjadi fokus dalam berbagai domain aplikasi. Memperhatikan penerapan YOLO dan Faster R-CNN dalam konteks Industri, terlihat bahwa YOLO lebih unggul dalam kecepatan, cocok untuk aplikasi *real-time* seperti mendeteksi cacat pada produk atau memantau keselamatan. Faster R-CNN lebih baik dalam kasus yang membutuhkan akurasi tinggi, seperti inspeksi visual otomatis atau analisis situasi rumit pada kendaraan otonom [5]. Studi yang menganalisis lima algoritma deteksi objek termasuk YOLO, SSD, dan Faster R-CNN di berbagai bidang seperti surveilans drone, deteksi lalu lintas, dan medis, juga mengkonfirmasi *trade-off* ini, model *one-stage* seperti YOLO lebih cepat dan cocok untuk aplikasi *real-time* untuk drone, sedangkan model *two-stage* layaknya Faster R-CNN lebih akurat dan menjadi pilihan utama di bidang medis yang menuntut presisi tinggi [6]. Pada kasus pengidentifikasi pil secara *real-time* di lingkungan farmasi, meskipun Faster R-CNN sedikit lebih akurat, YOLOv3 tetap direkomendasikan karena keseimbangan antara kecepatan inferensi yang tinggi memenuhi syarat *real-time* dan akurasi yang cukup baik, serta ukuran model yang lebih kecil [7]. Studi perbandingan sebelumnya dengan jumlah dataset 3908 pada domain deteksi kendaraan menggunakan citra dari kamera CCTV, diketahui bahwa terdapat *trade-off* antara akurasi dan kecepatan. hasil menunjukkan bahwa Faster R-CNN tetap lebih baik dalam mengklasifikasikan objek, terutama kendaraan yang ukurannya lebih kecil atau tumpang tindih. sementara itu, YOLOv8 lebih cepat dalam proses pengambilan keputusan meskipun sedikit kalah dalam akurasi untuk beberapa jenis kendaraan tertentu [8]. Temuan serupa yang semakin memperkuat adanya pertukaran antara akurasi dan kecepatan dalam perbandingan antara YOLOv5 dan Faster R-CNN. Dalam studi mereka, Faster R-CNN kembali menunjukkan keunggulan dalam hal akurasi (*mean Average Precision*),

meskipun selisihnya tipis, dengan mencapai skor 0.713 dibandingkan 0.682 untuk YOLOv5 [9].

Arsitektur *single-stage* seperti YOLO dan *two-stage* seperti Faster R-CNN menggunakan mekanisme deteksi yang berbeda. *Single-stage* mengutamakan kecepatan melalui prediksi langsung, sedangkan *two-stage* menonjol dalam akurasi melalui tahap proposal wilayah. Perbedaan ini menghasilkan keunggulan masing-masing sebagaimana ditunjukkan pada berbagai penelitian [8], [9], [10], sehingga pemilihan model bergantung pada konteks aplikasi dan karakteristik data. Untuk konteks sistem pemantauan gizi berbasis aplikasi *mobile*, tantangan ini menjadi semakin kompleks. Aplikasi semacam ini menuntut akurasi deteksi yang tinggi untuk memastikan estimasi kandungan nutrisi dan kalori seakurat mungkin. Aplikasi *mobile* bekerja dalam lingkungan dengan keterbatasan sumber daya komputasi dan membutuhkan kecepatan pengolahan yang tinggi untuk pengalaman pengguna yang responsif, serta ukuran model yang efisien [11]. Pada saat ini, masih terbatas penelitian yang secara langsung membandingkan kinerja generasi terbaru dari kedua arsitektur ini khususnya YOLOv8 dan Faster R-CNN pada domain spesifik makanan khas Indonesia. Domain ini menghadirkan tantangan karena tingginya variasi visual, kemiripan antar hidangan, dan kompleksitas penyajian. Oleh karena itu, penelitian ini bertujuan secara spesifik untuk mengisi celah pengetahuan tersebut. Dengan melakukan evaluasi komparatif yang mendalam dan sistematis antara YOLOv8 dan Faster R-CNN menggunakan dataset makanan Indonesia yang representatif, penelitian ini akan menentukan arsitektur mana yang mampu memberikan keseimbangan performa terbaik. Analisis akan berfokus pada metrik kuantitatif utama seperti akurasi (mAP), kecepatan inferensi (ms), dan ukuran model (MB), serta melibatkan analisis kualitatif untuk memahami bias dan mode kegagalan spesifik. Hasil evaluasi komprehensif ini diharapkan dapat memberikan rekomendasi teknis yang solid dan berbasis bukti empiris mengenai arsitektur yang paling optimal untuk menjadi fondasi pengembangan sistem pemantauan gizi berbasis *mobile* yang efisien, cepat, akurat, dan andal di masa depan, khususnya dalam konteks kuliner Indonesia.

II. METODE

Penelitian ini menggunakan beberapa tahapan untuk membandingkan kinerja dua arsitektur deteksi objek, YOLOv8 dan Faster R-CNN. Alur penelitian dirancang secara sistematis yang terdiri dari beberapa tahapan utama, sebagaimana diilustrasikan pada Gambar 1. Tahapan tersebut dimulai dari pengumpulan dataset, pemisahan data, prapemrosesan, penyesuaian format, pelatihan model, evaluasi, hingga analisis perbandingan kinerja.



Gambar 1. Alur Penelitian

A. Pengumpulan Dataset

Dataset penelitian ini dikumpulkan sepenuhnya melalui platform manajemen data Roboflow, bersumber dari repositori publik Roboflow Universe yang dikurasi khusus untuk domain makanan Indonesia. Dataset terdiri dari 1.325 citra digital berformat JPG yang dikelompokkan ke dalam 13 kelas kategori. Untuk menjamin representasi kondisi penggunaan nyata, dataset dipilih dengan variasi visual yang luas, mencakup perbedaan intensitas pencahayaan dan kompleksitas latar belakang. Seluruh proses kendali kualitas, mulai dari validasi presisi anotasi (*tight box*) hingga eliminasi citra buram, dilakukan menggunakan fitur kurasi terintegrasi Roboflow.

B. Splitting Data

Splitting data merupakan tahapan membagi dataset menjadi beberapa subset yang berbeda untuk keperluan pelatihan, validasi, dan pengujian model. Dalam penelitian ini dataset terkumpul dengan total 1325 seperti yang disajikan pada tabel 1. Data dibagi menjadi tiga bagian dengan rasio 70% untuk data latih (*training set*) sebanyak 933 citra, 20% untuk data validasi (*validation set*) sebanyak 261 citra, dan 10% untuk data uji (*test set*) sebanyak 131 citra. Pembagian dengan rasio ini merupakan praktik standar dalam *machine learning* yang bertujuan untuk memastikan proses pengembangan model berjalan dengan baik [12]. Adanya variasi jumlah sampel antar kelas yang signifikan, pembagian data dilakukan menggunakan teknik *Stratified Random Sampling* melalui platform Roboflow. Teknik ini memastikan bahwa proporsi setiap kategori makanan terdistribusi secara merata di seluruh subset, mencegah hilangnya representasi kelas minoritas pada data uji [13]. *Training set* digunakan sepenuhnya untuk melatih model mengenali pola visual,

sementara *validation set* berfungsi untuk memantau performa model selama pelatihan dan mencegah *overfitting*. *Test set* disimpan secara terpisah dan tidak pernah dilibatkan dalam proses pelatihan, sehingga mampu memberikan evaluasi yang objektif dan tidak bias terhadap kemampuan generalisasi model pada tahap akhir..

TABEL I
PEMBAGIAN DATA

No	Folder	Jumlah Data	Rasio
1	Train	933	70%
2	Valid	261	20%
3	Test	131	10%
4	Total	1325	100%

C. Pre-Processing Data

Preprocessing data adalah langkah penting untuk memastikan data dalam kondisi yang siap digunakan sebelum dimasukkan ke dalam model deteksi objek. Data yang sudah diproses secara rapi dan teratur biasanya berjalan lebih baik [14]. Dalam penelitian ini, seluruh citra diproses melalui standarisasi untuk menjamin keseragaman perlakuan pada kedua arsitektur model. Langkah awal melibatkan koreksi orientasi otomatis (*Auto-Orient*) untuk menghapus metadata rotasi dan memastikan seluruh citra berada dalam posisi tegak, mencegah distorsi semantik pada objek makanan. Penelitian ini menerapkan teknik *letterbox resizing*. Citra diubah ukurannya menjadi resolusi standar 640×640 piksel dengan tetap mempertahankan rasio aspek asli (*aspect ratio*), di mana area kosong yang tersisa diisi dengan *zero-padding*. Ukuran ini dipilih karena menawarkan keseimbangan optimal antara preservasi detail visual dan efisiensi beban komputasi [15]. Nilai intensitas piksel yang berada dalam rentang [0, 255] dinormalisasi menjadi [0, 1] agar data input memiliki distribusi yang standar, sehingga mempercepat kestabilan gradien selama proses pelatihan.

D. Penyesuaian Format Dataset

Tahap penyesuaian format dataset pada penelitian ini dipisah dan diekspor ke dalam dua format anotasi yang berbeda untuk memenuhi kebutuhan spesifik dari masing-masing *framework* dan arsitektur model. Untuk pelatihan model YOLOv8, anotasi diekspor dalam format YOLO TXT. Format ini sangat efisien untuk *framework* ultralytics karena berisi koordinat *bounding box* yang dinormalisasi, mempercepat proses pemuatannya [16]. Untuk pelatihan model Faster R-CNN, anotasi diekspor dalam format COCO JSON. Format ini lebih terstruktur dan kaya akan metadata, yang diperlukan oleh *framework* PyTorch untuk memahami relasi antara citra, kategori, dan anotasi, dikarenakan Faster R-CNN merupakan model deteksi objek dua tahap yang terdiri atas dua komponen utama, yaitu *Region Proposal Network* [17]. File COCO JSON ini kemudian divalidasi lebih lanjut menggunakan skrip kustom untuk memastikan integritasnya.

E. Pelatihan Model

Proses pelatihan model dirancang dengan perbandingan yang adil. Model YOLOv8 dilatih menggunakan kerangka kerja Ultralytics yang didasarkan pada bobot awal pre-trained yaitu `yolov8s.pt`, sedangkan Faster R-CNN dibangun menggunakan PyTorch dengan backbone ResNet-50 FPN. Penelitian ini menjaga kesamaan pada parameter iterasi utama, di mana kedua model dilatih selama 100 *epoch* menggunakan optimizer *Stochastic Gradient Descent* (SGD) agar mekanisme pembaruan bobot tetap sama.

Fitur augmentasi yang digunakan oleh kedua model adalah augmentasi geometri berupa *flip horizontal* dengan probabilitas 0.5. Untuk menghindari overfitting, diterapkan mekanisme Early Stopping dengan batas kesabaran 15 *epoch*. Pelatihan akan dihentikan otomatis jika tidak ada peningkatan kinerja. Penentuan model terbaik tidak berdasarkan hasil akhir iterasi, melainkan menggunakan metode Save Best Model, yaitu menyimpan checkpoint yang memiliki nilai mAP tertinggi pada data validasi untuk memastikan model memiliki kemampuan generalisasi yang baik.

F. Evaluasi Model

Setelah model terbaik dari masing-masing arsitektur yang dipilih berdasarkan performa puncak pada *validation set* berhasil diidentifikasi dan disimpan, tahap selanjutnya adalah evaluasi kinerja. Evaluasi ini dilakukan menggunakan data uji, yaitu subset data yang sepenuhnya terpisah dan tidak pernah digunakan selama proses pelatihan maupun validasi. Penggunaan *test set* yang independen ini bertujuan untuk mendapatkan estimasi yang objektif dan tidak bias mengenai kemampuan generalisasi model, yaitu seberapa baik model dapat beradaptasi dan membuat prediksi akurat pada data baru yang belum pernah dilihat sebelumnya. Penilaian terhadap kemampuan model dalam mendeteksi dan mengklasifikasikan 13 kelas makanan Indonesia dilakukan secara komprehensif dengan menghitung serangkaian metrik evaluasi. Metrik utama yang digunakan dalam penelitian ini adalah mean Average Precision (mAP) pada ambang batas mAP@0.50:0.95 untuk menilai kekuatan bounding box pada berbagai ambang batas, melengkapi metrik standar mAP@0.50. Metrik ini dipilih karena merupakan standar umum dalam evaluasi deteksi objek yang mengukur rata-rata presisi pada berbagai tingkat *recall*, dan memberikan gambaran yang baik mengenai kemampuan model dalam mengenali dan melokalisasi objek secara umum [18]. Selain akurasi, aspek efisiensi komputasi juga dievaluasi dengan mengukur kecepatan inferensi rata-rata per citra, dan ukuran file model, yang keduanya merupakan faktor penting untuk kelayakan implementasi pada aplikasi *mobile*.

$$mAP = \frac{1}{N} \sum_{i=1}^N AveragePrecision_i$$

G. Analisis Perbandingan Kinerja

Tahap terakhir dalam metodologi penelitian ini adalah analisis perbandingan kinerja yang komprehensif antara model YOLOv8 dan Faster R-CNN. Analisis ini bertujuan untuk menginterpretasikan hasil evaluasi secara holistik, melampaui perbandingan akurasi permukaan semata guna menarik kesimpulan yang valid untuk studi kasus deteksi makanan Indonesia. Fokus utama analisis diletakkan pada identifikasi *trade-off* arsitektural antara presisi lokalisasi ketat yang diukur dengan mAP@0.50:0.95, *recall*, dan keseimbangan harmonis antara keduanya yang direpresentasikan oleh *F1-Score*, serta efisiensi komputasi yang mencakup kecepatan inferensi dan ukuran model. Selain metrik kuantitatif, analisis kualitatif melalui visualisasi *Confusion Matrix* juga dilakukan untuk membedah pola kegagalan spesifik, seperti kecenderungan kesalahan klasifikasi antar-kelas atau kegagalan deteksi pada kelas minoritas akibat ketidakseimbangan data. Hasil dari seluruh temuan ini digunakan untuk menentukan arsitektur mana yang menawarkan keseimbangan paling optimal antara ketelitian dan responsivitas sebagai fondasi teknis pengembangan sistem pemantauan gizi berbasis *mobile*.

III. HASIL DAN PEMBAHASAN

Pada bagian ini, akan dipaparkan dan dianalisis secara komprehensif hasil dari serangkaian eksperimen yang dilakukan untuk mengevaluasi dan membandingkan kinerja model YOLOv8 dan Faster R-CNN dalam tugas deteksi makanan Indonesia. Sumber data pada penelitian ini diperoleh dari platform Roboflow, yang terdiri atas 13 kategori makanan Indonesia, yaitu Ayam Geprek, Ayam Goreng, Bakso, Ikan Bakar, Mie Goreng, Nasi Goreng, Nasi Kuning, Nasi Putih, Pecel, Rendang, Sate Ayam, Soto Ayam, dan Telur Goreng. Secara keseluruhan, jumlah citra unik yang tersedia sebanyak 1.325, dengan total 1.717 anotasi objek. Rincian distribusi anotasi pada setiap kelas disajikan pada Tabel 2.

TABEL II
JUMLAH ANOTASI KELAS

No	Kelas	Anotasi
1	Ayam Geprek	63
2	Ayam Goreng	404
3	Bakso	60
4	Ikan Bakar	47
5	Mie Goreng	103
6	Nasi Goreng	362
7	Nasi Kuning	46
8	Nasi Putih	147
9	Pecel	44
10	Rendang	57
11	Sate Ayam	67
12	Soto Ayam	156
13	Telur Goreng	161

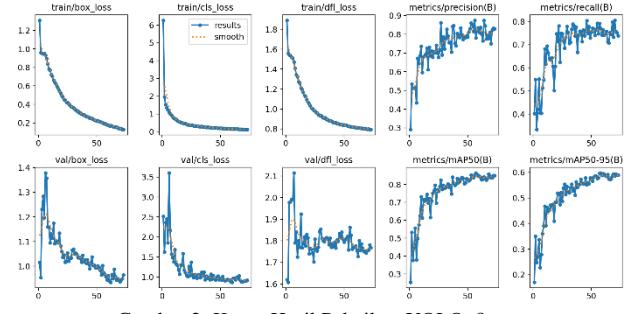
A. Pelatihan Model

Pelatihan model dilakukan secara terpisah untuk YOLOv8 dan Faster R-CNN menggunakan dataset latih dan divalidasi pada setiap *epoch* menggunakan dataset validasi. Tabel 3 menyajikan ringkasan hyperparameter utama yang digunakan dalam proses pelatihan kedua model. Parameter meliputi jumlah *epoch* maksimal, jenis *optimizer*, dan strategi augmentasi diatur secara seragam. Hal ini dilakukan agar perbedaan performa yang muncul hanya mencerminkan sifat dari struktur model itu sendiri, bukan karena pengaruh konfigurasi hiperparameter yang tidak konsisten.

TABEL III
HYPERPARAMETER PELATIHAN MODEL

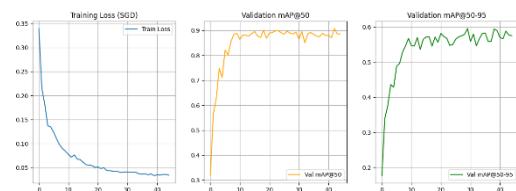
No	Hyperparameter	YOLOv8	Faster R-CNN
1	Arsitektur Dasar	yolov8s.pt	ResNet-50 FPN
2	Framework	Ultralytics	PyTorch
3	Optimizer	SGD	SGD
4	Max Epoch	100	100
5	Early Stopping	Patience = 15 Epochs	Patience = 15 Epochs
6	Batch Size	16	4
7	Input Size	640×640	640×640
8	Learning Rate	0.01	0.005
9	Augmentasi	Flip Horizontal	Flip Horizontal

Pelatihan YOLOv8 dikonfigurasi dengan batas maksimum 100 *epoch*. Gambar 2 menyajikan visualisasi kurva *loss* dan metrik performa yang dihasilkan. Berdasarkan grafik *training loss*, ketiga komponen utama yaitu *box_loss*, *cls_loss*, dan *dfl_loss* menunjukkan tren penurunan yang konsisten dan stabil, mengindikasikan bahwa model belajar memperbaiki prediksi *bounding box* dan klasifikasi secara efektif tanpa fluktuasi gradien yang ekstrem. Grafik *validation loss* juga menunjukkan tren menurun yang kemudian melandai secara alami tanpa adanya divergensi signifikan, menandakan bahwa model memiliki kemampuan generalisasi yang baik dan terhindar dari *overfitting*. Metrik performa pada *validation set*, khususnya mAP@50, menunjukkan peningkatan progresif yang pesat pada 20 *epoch* awal sebelum mencapai fase stabilitas (*plateau*). Mekanisme *Early Stopping* terbukti bekerja efektif dengan menghentikan pelatihan secara otomatis pada *epoch* ke-74, karena tidak ada peningkatan kinerja yang signifikan selama 15 *epoch* berturut-turut. Model terbaik (*best.pt*) kemudian dipilih berdasarkan *checkpoint* yang mencapai nilai mAP validasi tertinggi sebelum pelatihan dihentikan.



Gambar 2. Kurva Hasil Pelatihan YOLOv8

Pelatihan Faster R-CNN dikonfigurasi menggunakan optimizer *Stochastic Gradient Descent* (SGD) dengan batas maksimum 100 *epoch*. Gambar 3 menunjukkan bagaimana kurva *loss* dan metrik validasi berubah selama proses pelatihan. Kurva *Training Loss (SGD)* menunjukkan pola penurunan yang halus dan stabil, menandakan proses pembaruan bobot yang terkendali tanpa fluktuasi ekstrem. Pada kurva validasi, metrik mAP@50 mengalami peningkatan pesat dan mencapai fase stabilitas (*plateau*) di kisaran 0.89 mulai *epoch* ke-15. Sementara itu, kurva mAP@50-95 terus menunjukkan tren peningkatan bertahap hingga *epoch* ke-30, mengindikasikan model terus memperbaiki presisi lokalisasi objek seiring bertambahnya iterasi. Mekanisme *Early Stopping* bekerja efektif dengan menghentikan pelatihan pada *epoch* ke-45, dikarenakan tidak terdeteksi peningkatan kinerja signifikan pada metrik mAP@50-95 selama 15 *epoch* berturut-turut. Model terbaik kemudian disimpan berdasarkan *checkpoint* dengan performa puncak sebelum penghentian tersebut.



Gambar 3. Kurva Hasil Pelatihan Faster R-CNN

B. Evaluasi Model

Data pada Tabel 4 menunjukkan hasil perbandingan pada data uji. Dari segi akurasi, YOLOv8 mencatatkan mAP@50 sebesar 88.6% dan mAP@50-95 sebesar 62.0%, mengungguli Faster R-CNN yang masing-masing mencapai 85.5% dan 55.6%. Keunggulan ini mengindikasikan bahwa arsitektur *one-stage* modern dengan pendekatan *anchor-free* mampu mempelajari fitur visual makanan Indonesia yang kompleks dengan lebih efektif dibandingkan *backbone* ResNet-50 pada Faster R-CNN, bahkan pada kriteria lokalisasi ketat (IoU 0.50-0.95). Perbedaan hasil paling mencolok terlihat pada metrik *Recall*. YOLOv8 mencapai skor 87.7%, jauh melampaui Faster R-CNN yang tertahan di angka 61.7%. Selisih 26% ini menunjukkan bahwa Faster R-CNN memiliki kelemahan signifikan dalam mendeteksi objek atau *False Negative* tinggi, kemungkinan akibat kegagalan *Region*

Proposal Network (RPN) dalam menghasilkan kandidat area yang akurat pada objek dengan variasi intra-kelas tinggi. Kesenjangan dalam *Recall* berdampak langsung pada keseimbangan performa global yang diukur melalui *F1-Score*. YOLOv8 mencapai skor F1 sebesar 0.84, jauh lebih baik dibandingkan Faster R-CNN yang hanya mencapai 0.72. Angka ini menunjukkan bahwa YOLOv8 mampu menjaga keseimbangan yang baik antara akurasi dan tingkat deteksi yang lengkap. Keunggulan YOLOv8 juga terlihat di bidang efisiensi. Model ini bekerja dengan kecepatan inferensi rata-rata 13.5 ms per gambar, sekitar 7.7 kali lebih cepat, dan ukurannya hanya 21.5 MB, atau 7.3 kali lebih ringan dibandingkan Faster R-CNN yang membutuhkan waktu 104.8 ms dan ukuran file sebesar 158.3 MB.

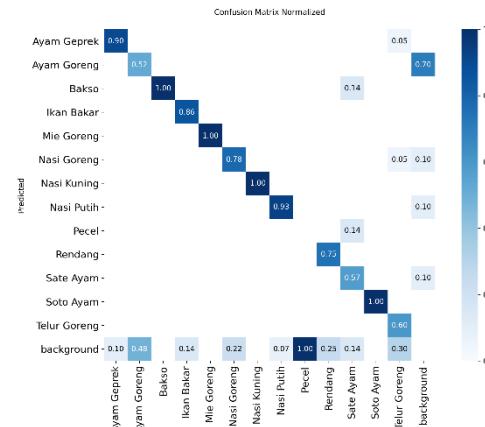
TABEL IV

HASIL EVALUASI KINERJA PADA TEST SET

NO	Metrik Evaluasi	YOLOv8	Faster R-CNN
1	mAP50	0.886	0.855
2	mAP@50-95	0.620	0.556
3	Recall	0.877	0.617
4	F1-Score	0.84	0.72
5	Inferensi	13.5 ms	104.8 ms
6	Ukuran File Model	21.5 MB	158.3 MB

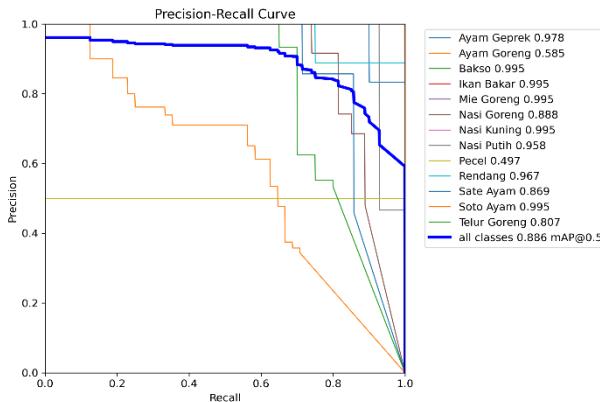
Dari analisis *Confusion Matrix Normalized* pada Gambar 4 memperlihatkan karakteristik deteksi model YOLOv8 yang sangat kontras. Pada satu sisi, model menunjukkan keunggulan performa pada kelas dengan fitur visual yang distingatif, di mana 'Bakso', 'Mie Goreng', 'Nasi Kuning', dan 'Soto Ayam' mencapai skor 1.00. Kemampuan model dalam mengenali berbagai jenis makanan juga cukup baik, terlihat dari skor 'Nasi Putih' yang mendapat 0.93 dan 'Ayam Geprek' 0.90.

Matriks ini juga menunjukkan masalah pada kelas 'Pecel', yang secara visual sepenuhnya dikategorikan sebagai *background*. Hal ini terjadi karena *confidence score* deteksi model pada sampel tersebut berada di bawah ambang batas visualisasi standar, meskipun secara statistik objek tersebut berhasil terdeteksi dengan kemungkinan yang rendah. Tantangan yang dihadapi oleh arsitektur *single-stage* terlihat pada kelas 'Ayam Goreng' dan 'Telur Goreng', yang masing-masing memiliki tingkat kesalahan *False Negative* sebesar 48% dan 30%. Tingginya angka pada baris *background* ini mengindikasikan bahwa model masih menghadapi kesulitan dalam memisahkan objek gorengan yang memiliki tekstur kompleks dari latar belakang, terutama ketika objek tersebut memiliki kemiripan warna yang tinggi dengan wadah atau lingkungan sekitarnya.



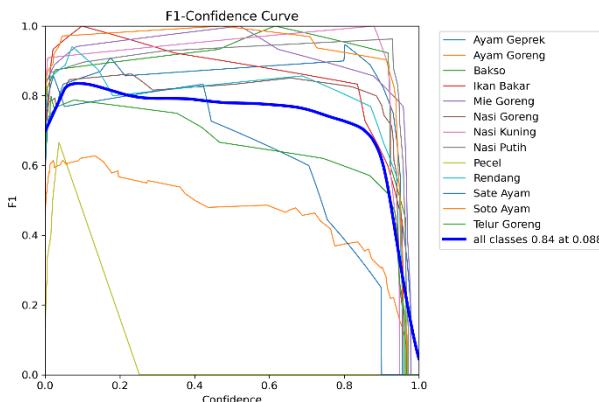
Gambar 4. Confusion Matrix Normalized

Kurva *Precision-Recall* yang terlihat pada Gambar 5 menunjukkan bagaimana model YOLOv8 bekerja dalam mendeteksi objek, dengan ratarata presisi dan *recall* yang cukup baik. Pada ambang batas IoU 0.5, model mencapai nilai *mean Average Precision* (mAP) sebesar 0.886 atau 88.6%. Kurva rata-rata untuk semua kelas menunjukkan bahwa model bisa menjaga tingkat presisi yang tinggi hingga *recall* mencapai 0.8, lalu mengalami penurunan, yang menunjukkan bahwa model memiliki performa yang stabil secara umum. Model menunjukkan performa yang sangat baik pada lima kelas, yaitu "Bakso", "Ikan Bakar", "Mie Goreng", "Nasi Kuning", dan "Soto Ayam". Semua kelas ini mencapai skor yang sangat tinggi, yaitu sekitar 0.995, dengan kurva yang berada di sudut kanan atas, menunjukkan tingkat deteksi yang sangat akurat. Grafik ini menunjukkan ketidakseimbangan dalam kinerja pada beberapa kelas. Kelas yang paling rendah adalah "Pecel", yang memiliki skor terendah sebesar 0.497, dengan kurva yang datar, menunjukkan keterbatasan dalam jumlah data uji. Selain itu, kelas "Ayam Goreng" menunjukkan penurunan signifikan dalam performa, yaitu skor sebesar 0.585. Kurva untuk kelas ini melandai dengan tajam sejak awal, menunjukkan kesulitan model dalam menjaga presisi ketika meningkatkan sensitivitas deteksi terhadap objek gorengan. Kelas lainnya, seperti "Telur Goreng", berada sedikit di bawah rata-rata global dengan skor sebesar 0.807.



Gambar 5. Kurva Precision-Recall YOLOv8

Kurva *F1-Confidence* yang disajikan pada Gambar 6 memetakan hubungan antara *F1-Score* dengan ambang batas kepercayaan model. Analisis ini menentukan titik operasional di mana keseimbangan antara *Precision* dan *Recall* mencapai nilai maksimal. Berdasarkan grafik, *F1-Score* rata-rata puncak untuk seluruh kelas tercatat sebesar 0.84, yang dicapai pada *confidence threshold* 0.088. Nilai *threshold* optimal yang rendah ini mengindikasikan bahwa model beroperasi paling efektif saat disetel lebih sensitif untuk meminimalkan *False Negative*, strategi yang diperlukan untuk mengakomodasi kelas-kelas sulit seperti 'Pecel' dan 'Ayam Goreng' yang performanya menurun drastis saat ambang batas kepercayaan dinaikkan. Sebagai perbandingan signifikan, arsitektur Faster R-CNN hanya mencatatkan estimasi *F1-Score* sebesar 0.72. Kesenjangan ini menegaskan superioritas YOLOv8 dalam mempertahankan keseimbangan harmonis antara ketepatan dan kelengkapan deteksi, kontras dengan Faster R-CNN yang kinerjanya terdegradasi akibat rendahnya sensitivitas.



Gambar 6. F1-Confidence YOLOv8

C. Perbandingan Kinerja

Hasil evaluasi eksperimen ini membuktikan bahwa YOLOv8 mengungguli kinerja di seluruh parameter evaluasi utama. Pada aspek akurasi deteksi, YOLOv8 mencapai skor mAP@0.50 sebesar 88.6%, mengungguli Faster R-CNN yang

mencatatkan 85.5%. Keunggulan ini semakin kuat pada metrik ketelitian lokalisasi ketat mAP@0.50-95, di mana YOLOv8 memimpin dengan skor 62.0% berbanding 55.6%. Hal ini mengindikasikan bahwa arsitektur *single-stage* modern dengan *backbone* CSPDarknet mampu mempelajari fitur visual makanan yang kompleks dengan lebih efektif dibandingkan arsitektur *two-stage* ResNet-50. Perbedaan utama dalam kemampuan model terletak pada sensitivitasnya dalam mendekripsi objek. YOLOv8 memiliki tingkat *Recall* yang sangat baik yaitu 87,7%, jauh lebih baik dibandingkan Faster R-CNN yang hanya mencapai 61,7%. Perbedaan 26% ini menunjukkan bahwa YOLOv8 lebih baik dalam mencegah kesalahan deteksi yang mengakibatkan objek *False Negative*, yang merupakan faktor penting dalam keandalan aplikasi estimasi nutrisi. Dampak dari keunggulan sensitivitas ini terakumulasi pada metrik *F1-Score*, yang merepresentasikan keseimbangan harmonis antara presisi dan *recall*. YOLOv8 berhasil mencatatkan *F1-Score* puncak sebesar 0.84, unggul signifikan dibandingkan Faster R-CNN yang hanya mencapai 0.72. Angka ini mengonfirmasi bahwa YOLOv8 menawarkan kompromi kinerja yang paling optimal, mampu mendekripsi objek secara lengkap tanpa mengorbankan ketepatan prediksi.

Dari perspektif efisiensi komputasi, keunggulan YOLOv8 semakin kuat. Model ini mampu memproses citra dengan kecepatan inferensi rata-rata 13.5 ms, sekitar 7.7 kali lebih cepat dibandingkan Faster R-CNN yang mencapai 104.8 ms. Ukuran file model yang dihasilkan YOLOv8 hanya 21.5 MB, atau 7.3 kali lebih ringan dibandingkan kompetitornya yang berukuran 158.3 MB. Kombinasi antara akurasi, sensitivitas tinggi, dan efisiensi sumber daya, menjadikan YOLOv8 pilihan yang paling optimal dan praktis untuk implementasi sistem deteksi makanan pada perangkat *mobile*.

IV. KESIMPULAN

Hasil dari penelitian pada dataset makanan Indonesia, dapat disimpulkan bahwa arsitektur YOLOv8 menunjukkan performa yang relatif lebih sesuai untuk kebutuhan sistem pemantauan gizi dibandingkan Faster R-CNN pada konfigurasi dan kondisi eksperimen yang digunakan. Model YOLOv8 mencatatkan mAP50 sebesar 0.886, mAP@50-95 0.620, *Recall* 0.877, dan *F1-Score* 0.84. Faster R-CNN memperoleh mAP50 sebesar 0.855, mAP@50-95 0.556, *Recall* 0.617, dan *F1-Score* 0.72. Hasil ini menunjukkan bahwa meskipun Faster R-CNN mampu mempertahankan presisi yang baik, sensitivitasnya yang lebih rendah menyebabkan jumlah *False Negative* lebih tinggi, sehingga cakupan deteksinya kurang lengkap dibandingkan YOLOv8 dalam konteks dataset penelitian ini.

Dari aspek efisiensi komputasi, YOLOv8 mencatatkan waktu inferensi rata-rata 13.5 ms dengan ukuran model 21.5 MB, sedangkan Faster R-CNN membutuhkan 104.8 ms dengan ukuran model 158.3 MB. Hal ini mengindikasikan bahwa YOLOv8 jauh lebih ringan dan responsif untuk

digunakan pada perangkat dengan sumber daya terbatas. Rekomendasi penggunaan YOLOv8 tetap bersifat kondisional, karena dibatasi oleh ukuran dataset, ketidakseimbangan kelas, serta belum adanya pengujian langsung pada perangkat *mobile*.

Sebagai upaya penyempurnaan sistem di masa depan, disarankan untuk melakukan penyeimbangan dataset dengan menambah variasi sampel pada kelas minoritas guna meningkatkan stabilitas deteksi. Selain itu, eksplorasi terhadap varian model yang lebih ringan seperti YOLOv8-Nano atau penerapan teknik kuantisasi direkomendasikan untuk memaksimalkan kompatibilitas pada perangkat seluler berspesifikasi rendah. Pengembangan lebih lanjut juga perlu diarahkan pada integrasi modul estimasi volume dan basis data nutrisi untuk melengkapi fungsionalitas sistem pemantauan gizi.

DAFTAR PUSTAKA

- [1] C. Serón-Arbeloa *et al.*, “Malnutrition Screening and Assessment,” *Nutrients*, vol. 14, no. 12, Jun. 2022, doi: 10.3390/nu14122392.
- [2] M. A. A. Al-qaness, A. A. Abbasi, H. Fan, R. A. Ibrahim, S. H. Alsamhi, and A. Hawbani, “An improved YOLO-based road traffic monitoring system,” *Computing*, vol. 103, no. 2, pp. 211–230, Feb. 2021, doi: 10.1007/s00607-020-00869-8.
- [3] G. Cui, S. Wang, Y. Wang, Z. Liu, Y. Yuan, and Q. Wang, “Preceding vehicle detection using faster R-CNN based on speed classification random anchor and Q-square penalty coefficient,” *Electronics (Switzerland)*, vol. 8, no. 9, Sep. 2019, doi: 10.3390/electronics8091024.
- [4] S. S. Sentik and S. Jakarta, “Penerapan Algoritma You Only Look Once v5 Untuk Deteksi Kualitas Buah Alpukat Menggunakan Python Berbasis Web,” 2024.
- [5] N. Rane, “YOLO and Faster R-CNN object detection for smart Industry 4.0 and Industry 5.0: applications, challenges, and opportunities,” *SSRN Electronic Journal*, 2023, doi: 10.2139/ssrn.4624206.
- [6] O. Olorunshola, P. Jemitolu, and A. Ademuwagun, “Comparative Study of Some Deep Learning Object Detection Algorithms: R-CNN, FAST R-CNN, FASTER R-CNN, SSD, and YOLO,” *Nile Journal of Engineering and Applied Science*, no. 0, p. 1, 2023, doi: 10.5455/njeas.150264.
- [7] L. Tan, T. Huangfu, L. Wu, and W. Chen, “Comparison of YOLO v3, Faster R-CNN, and SSD for Real-Time Pill Identification,” Jul. 30, 2021, doi: 10.21203/rs.3.rs-668895/v1.
- [8] P. Akurası *et al.*, “Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi) 2025 388,” Online.
- [9] Z. Guo, C. Wang, G. Yang, Huang, and G. Li, “MSFT-YOLO: Improved YOLOv5 Based on Transformer for Detecting Defects of Steel Surface,” *Sensors*, vol. 22, no. 9, May 2022, doi: 10.3390/s22093467.
- [10] W. Liu, G. Ren, R. Yu, S. Guo, J. Zhu, and L. Zhang, “Image-Adaptive YOLO for Object Detection in Adverse Weather Conditions,” 2022. [Online]. Available: www.aaai.org
- [11] A. Körez and N. Barişçi, “Object detection with low capacity GPU systems using improved faster R-CNN,” *Applied Sciences (Switzerland)*, vol. 10, no. 1, Jan. 2020, doi: 10.3390/app10010083.
- [12] 2019 IEEE 5th International Conference for Convergence in Technology (I2CT). IEEE, 2019.
- [13] R. Iliyasu and I. Etikan, “Comparison of quota sampling and stratified random sampling,” *Biom Biostat Int J*, vol. 10, no. 1, pp. 24–27, Feb. 2021, doi: 10.15406/bbij.2021.10.00326.
- [14] M. Iksan Maulana and U. Hayati, “Perbandingan Algoritma Naïve Bayes Dan K-Nearest Neighbors Untuk Klasifikasi Topik Berita Pada Situs Detik.Com,” 2024.
- [15] H. Talebi, P. Milanfar, and G. Research, “Learning to Resize Images for Computer Vision Tasks.”
- [16] R. Khanam and M. Hussain, “What is YOLOv5: A deep look into the internal features of the popular object detector,” Jul. 2024, [Online]. Available: http://arxiv.org/abs/2407.20892
- [17] E. Malagoli and L. Di Persio, “2D Object Detection: A Survey,” Mar. 01, 2025, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/math13060893.
- [18] M. I. Pratama, N. Nurchim, and E. Purwanto, “Analisis Perbandingan Metode Yolo Dan Faster R-CNN Dalam Deteksi Objek Manusia,” *Progresif: Jurnal Ilmiah Komputer*, vol. 21, no. 2, p. 545, Aug. 2025, doi: 10.35889/progresif.v21i2.2890.