ANALISIS KUALITAS DATA CITRA DAN DAMPAKNYA TERHADAP AKURASI BOUNDING BOX PADA MODEL YOLO V8

Suprayogie[1], Ricky Perdana Kusuma[2], Heri Suhendar[3]
[1]STMIK LIKMI
Ir. H. Juanda No. 96 Bandung
[2]Universitas Catur Insan Cendekia
Jl. Kesambi No. 202, Drajat, Kec. Kesambi, Kota Cirebon, Jawa Barat 45133
[3]Institut Teknologi Garut
Jl. Mayor Syamsu No. 1, Jayaraga, Kec. Tarogong Kidul, Kabupaten Garut, Jawa Barat 44151
e-mail: 2025210034.student@likmi.ac.id, rickyperdanak@cic.ac.id[2], herysuhendar@itg.ac.id[3]

Abstrak

Kualitas data citra merupakan faktor krusial yang menentukan performa model deteksi objek berbasis deep learning, khususnya YOLOv8. Penelitian ini menganalisis pengaruh kualitas data citra terhadap akurasi bounding box pada model YOLOv8 menggunakan pendekatan eksperimental kuantitatif. Dataset yang digunakan terdiri dari 50 citra dengan variasi resolusi yang berbeda, yang dianotasi menggunakan platform Roboflow dan dilatih pada Google Colab selama 100 epoch. Kualitas citra diukur menggunakan empat parameter utama, yaitu resolusi (px), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), serta akurasi bounding box yang diukur melalui Intersection over Union (IoU) dan Mean Average Precision (mAP). Hasil analisis menunjukkan bahwa sebesar 66% citra memiliki resolusi tinggi (≥640px), 64% citra berada pada rentang PSNR 30–40 dB (kategori baik), dan 98% citra memiliki nilai SSIM pada rentang 0,75–1,0 (kemiripan struktural tinggi). Evaluasi model menghasilkan nilai Mean IoU sebesar 0,9156 dan mAP@0.5 sebesar 98,36%, serta mAP@0.5:0.95 sebesar 89,78%, yang keseluruhannya termasuk kategori sangat baik. Penelitian ini membuktikan bahwa kualitas data citra yang baik dari sisi resolusi, tingkat noise, dan konsistensi struktur visual berkontribusi signifikan terhadap peningkatan akurasi lokalisasi bounding box pada model YOLOv8.
Kata kunci: *YOLOv8, deteksi objek, bounding box, kualitas citra, mAP*

Abstract

Image data quality is a crucial factor determining the performance of deep learning-based object detection models, particularly YOLOv8. This study analyzed the effect of image data quality on bounding box accuracy in the YOLOv8 model using a quantitative experimental approach. The dataset consisted of 50 images with varying resolutions, annotated using the Roboflow platform and trained on Google Colab for 100 epochs. Image quality was measured using four main parameters: resolution (px), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and bounding box accuracy assessed through Intersection over Union (IoU) and Mean Average Precision (mAP). Results showed that 66% of images had high resolution (≥640px), 64% of images achieved PSNR values in the 30–40 dB range (good category), and 98% of images had SSIM values between 0.75 and 1.0 (high structural similarity). Model evaluation yielded a Mean IoU of 0.9156, mAP@0.5 of 98.36%, and mAP@0.5:0.95 of 89.78%, all classified as excellent. This study demonstrates that good image data quality in terms of resolution, noise level, and structural visual consistency significantly contributes to improved bounding box localization accuracy in the YOLOv8 model

Keywords*: YOLOv8, object detection, bounding box, image quality, mAP*

1. PENDAHULUAN

Dalam konteks deteksi objek berbasis deep learning, masalah kualitas data citra merupakan masalah yang sangat penting dan penting, tetapi sering kali kurang diperhatikan. Sistem deteksi objek seringkali bekerja dalam lingkungan yang tidak ideal di dunia nyata. Faktor-faktor seperti pencahayaan yang tidak memadai, resolusi kamera yang rendah, gangguan suara dari sensor, atau cuaca yang mengurangi ketajaman dan kejernihan gambar adalah beberapa contoh situasi yang tidak ideal. Kualitas citra—yang merupakan input untuk model algortima—dipengaruhi secara langsung oleh kondisi tersebut. Pada akhirnya, kondisi tersebut berdampak pada ketepatan deteksi dan ketepatan bounding box yang dihasilkan. Salah satu fungsi penting dalam computer vision adalah deteksi objek, yang bertujuan untuk secara otomatis mengidentifikasi dan melokalisasi objek dalam gambar atau video. Bidang deep learning telah mengalami kemajuan pesat, dan arsitektur Convolutional Neural Network (CNN) telah meningkatkan kinerja sistem deteksi objek kontemporer [8]. Model YOLO (You Only Look Once) adalah algoritma deteksi objek berbasis deep learning yang dapat mendeteksi objek secara real time dan memproses gambar dalam satu kali proses [11]. Ini menjadikannya algoritma yang paling banyak digunakan. Algoritma ini membagi gambar menjadi grid dan memprediksi bounding box dengan skor kepercayaan [11]. Kemudian, untuk melakukan seleksi terhadap deteksi yang tumpah tindih, algoritma ini menggunakan Non-Maximum Suppression . YOLO dapat mengidentifikasi objek dan menjelaskannya. Metode yang digunakan dapat melewatkan gambar melalui Neural Network, yang memungkinkan model mendeteksi semua objek secara bersamaan. YOLO menggunakan prediksi berbasis grid dan prediksi bounding box, dan model dapat mengetahui apakah objek berada di dalam bounding box. YOLO juga menggunakan probabilitas kelas untuk mengklasifikasikan dan menentukan objek, dan salah satu dari banyak keunggulan YOLO adalah sangat cepat dan dapat diterapkan pada banyak objek penelitian yang berbeda. Metode ini juga sangat akurat, terutama untuk objek alami. YOLO direkomendasikan karena memiliki kecepatan pemrosesan gambar 45 Frame per second (FPS) dan memiliki dua kali lipat Average Precesion (mAP) dibandingkan dengan sistem real time lainnya [11]. YOLO juga sangat akurat dengan sedikit kesalahan. Generalisasi dalam pembelajaran mesin berarti bahwa model tetap akurat saat diberi data baru yang berbeda dari data pelatihannya. YOLO memiliki kemampuan ini, dan versi YOLO yang lebih baru semakin mampu beradaptasi dengan lebih banyak objek. Analisis ini berfokus pada bounding box dan menyelidiki dampak kualitas gambar pada deteksi objek model YOLOv8.

2. METODE PENELITIAN

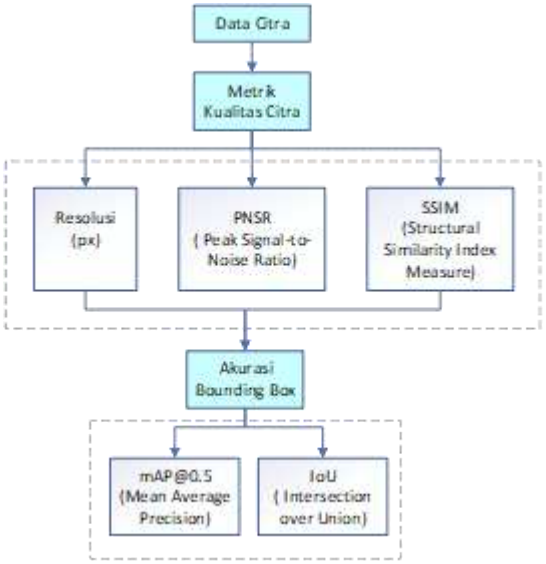
Untuk menganalisis hubungan antara kualitas gambar dan bounding box pada model YOLOv8, penulis menggunakan dataset gambar sebagai bahan analisis. Penelitian ini menggunakan pendekatan ekperimental kuantitatif. Tahapan awal untuk analisis kualitas data citra ini menggunakan dataset yang diolah menggunakan Roboflow, yang merupakan sebuah platform berbasis cloud yang menggabungkan berbagai tahapan dalam pengembangan model Computer Vision. Tahapan-tahapan yang ada di Roboflow meliputi pengumpulan data dan pengembangan model vision komputer.

Selama tahap anotasi, Roboflow menyediakan fitur label data seperti bounding box), polygon segmentation, keypoint annotation, dan classification label. Setelah anotasi selesai, dataset dapat diproses menggunakan fitur preprocessing seperti resize gambar gambar, normalisasi, auto orientation, dan filter kualitas gambar. Untuk membuat model lebih tangguh terhadap variasi kondisi lapangan maka roboflow dapat menggunakan augmentasi data seperti rotasi, flip, cropping, pengaturan kecerahan, blur.

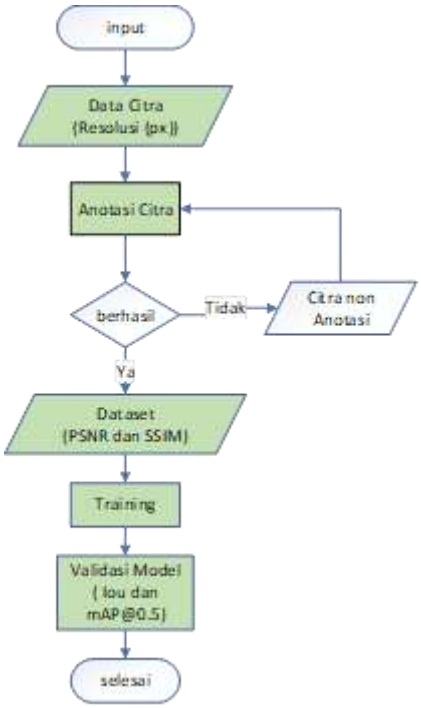
Sebelumnya, data citra gambar diunggah ke Roboflow untuk dilakukan anotasi menggunakan jenis bounding box. Setelah itu, dataset dapat diekspor secara instan dalam format yang kompatibel dengan YOLO, yang biasanya terdiri dari file gambar, label.txt, dan file konfigurasi seperti data yaml. Selain itu, key API dan script ekspor dari Roboflow memungkinkan dataset diunduh langsung ke Google Colab tanpa perlu melakukan download manual.

Google Colab, juga dikenal sebagai Google Colaboratory, adalah platform berbasis cloud yang dikembangkan oleh Google untuk menjalankan kode Python. Selanjutnya, Google Colab berfungsi sebagai environment Python berbasis cloud yang menyediakan GPU/TPU untuk menjalankan pelatihan dan evaluasi model. Dalam prosesnya, data citra yang telah dianotasi dari Roboflow dikirim ke Google Colab, dan kemudian model YOLOv8 dilatih. Model akan menghasilkan metrik evaluasi selama proses pelatihan; parameter metrik kualitas digunakan sebagai analisis yang berkaitan dengan bounding box.. Namun, untuk mengetahui akurasi bounding box, dapat dihitung mAP (Mean Average Precession) dan Iou (Intersection over Union). Tujuan utama penelitian ini adalah untuk mengevaluasi hubungan antara kualitas gambar

sebagai kumpulan data dan akurasi kotak batas pada model deteksi objek YOLOv8, sehingga penelitian ini dapat menghasilkan kriteria kulitas gambar yang diharapkan. Parameter kualitas data seperti resolusi (px), Peak Signal to Noise Ratio (PNSR), dan Structural Similiarity Index Measure (SSIM) dapat dilihat pada bagan di bawah ini.



Gambar 1. Bagan Parameter Kualitas Citra dan Akurasi Bounding Box

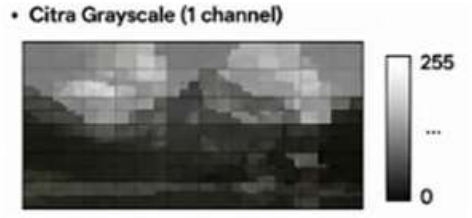


Gambar 2. Alur Proses Analisis Kualitas Citra

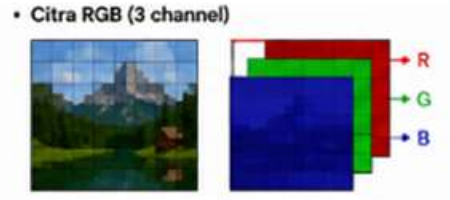
2.1 Pengumpulan Data Citra

Untuk deep learning, data Citra adalah fondasi utama yang menentukan kemampuan model untuk memahami dan mengidentifikasi pola visual. Ini adalah representasi visual dari suatu objek atau lingkungan yang direkam secara digital. Untuk memulai proses pelatihan model, kualitas data Citra sangat penting. Sebuah Citra digital terdiri dari piksel-piksel, masing-masing dengan nilai intensitas warna tertentu [6].

Dalam gambar grayscale, setiap piksel diwakili oleh satu nilai intensitas dalam rentang 0–255. Sebaliknya, dalam gambar berwarna model RGB, setiap piksel diwakili oleh tiga nilai yaitu merah, hijau, dan biru, masing-masing dalam rentang yang sama. Data gambar dalam penelitian deteksi objek berbasis YOLO biasanya dikumpulkan dari berbagai sumber, seperti dataset publik, kamera industri, atau kamera CCTV. Sebelum digunakan, citra harus melalui proses preprocessing. Ini termasuk memeriksa resolusi, mengubah format, normalisasi nilai piksel, dan menghilangkan gambar yang rusak atau tidak relevan. Keseimbangan jumlah sampel per kelas objek (class balance), keberagaman kondisi pencahayaan dan sudut pengambilan gambar, dan konsistensi format dan ukuran gambar adalah tiga komponen penting dalam pengelolaan data gambar yang secara langsung memengaruhi kemampuan generalisasi model yang dihasilkan. Pada peneitian ini penulis menggunakan 50 citra gambar dengan resolusi yang berbeda untuk setiap citra gambarnya.



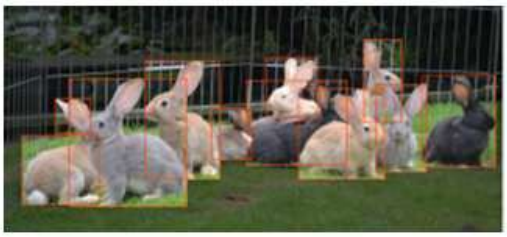
Gambar 3. Ilustrasi Citra Grasyscale



Gambar 4. Ilustrasi Citra RGB

2.2 Proses Anotasi Citra

Anotasi citra adalah proses pelabelan objek dalam gambar untuk memberikan informasi lokasi dan kategori objek kepada model yang akan dilatih. Dalam deteksi objek, anotasi ditampilkan dalam bentuk kotak bounding box, yaitu kotak persegi panjang yang mengelilingi objek target dengan koordinat yang tepat. Dalam proses anotasi citra, ada dua pendekatan utama yaitu anotasi manual dan anotasi otomatis. Dalam pendekatan manual, pengguna menggunakan antarmuka grafis yang tersedia di Roboflow untuk menggambar bounding box di setiap gambar. Meskipun membutuhkan banyak waktu dan energi, metode ini menghasilkan kebenaran dasar yang akurat. Anotasi otomatis membuat bounding box berdasarkan prompt teks atau deteksi visual dengan menggunakan model AI seperti Grounding DINO atau Segment Anything Model (SAM) [7]. Dalam hal waktu, metode ini jauh lebih efisien daripada metode lain, namun, akurasi anotasi sangat bergantung pada kualitas gambar yang dimasukkan. Anotasi otomatis akan menjadi tidak akurat jika gambarnya buram, beresolusi rendah, atau banyak noise. Dalam computer vision, Roboflow adalah platform yang populer untuk mengelola dataset. Pilihan anotasi manual berbasis web, anotasi otomatis menggunakan model fondasi, manajemen versi dataset, dan pipeline augmentasi data yang terintegrasi adalah semua fitur yang ditawarkan oleh platform ini. Roboflow membantu proses pembentukan dataset dan anotasi dalam alur penelitian ini. Kualitas anotasi, baik otomatis maupun manual, dipengaruhi oleh seberapa presisi batas bounding box ditentukan. Anotasi yang terlalu longgar (over annotation) atau terlalu ketat (under annotation) akan menghasilkan kebenaran dasar yang tidak akurat, yang pada akhirnya akan mengurangi kinerja model yang dilatih menggunakan data tersebut. Analisis ini akan menguji kualitas gambar dengan model YOLO.



Gambar 5. Anotasi Bounding Box



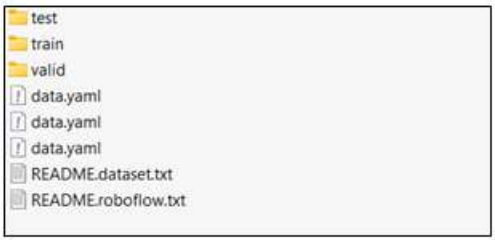
Gambar 6. Penamaan Kategori Kelas

2.3 Dataset untuk Model YOLOv8

Dataset dalam pengolahan citra digital dan computer vision adalah sekumpulan gambar atau video yang dilengkapi dengan informasi anotasi, yaitu label atau penanda yang menjelaskan konten visual dari setiap gambar. Dataset sangat penting untuk kualitas model yang dihasilkan dalam penelitian ilmiah, khususnya dalam bidang deteksi objek berbasis deep learning. Dataset hasil pengolahan dari Roboflow untuk model YOLOv8 terdiri dari dua bagian utama secara struktural. Folder pertama adalah gambar. Folder kedua adalah folder label, yang berisi file teks jenis.txt yang memiliki nama yang sama dengan file gambar yang relevan. Dalam file teks, setiap baris menunjukkan satu objek yang teranotasi dengan format nomor kelas cx, cy, w, h, di mana cx dan cy adalah koordinat pusat bounding box yang dinormalisasi terhadap lebar dan tinggi gambar, dan w dan h adalah bounding box lebar dan tinggi, yang masing-masing dinormalisasi ke dalam rentang nilai 0 hingga 1. File konfigurasi YAML (data.yaml) adalah komponen ketiga yang sangat penting yang berisi metadata dataset. File ini mencakup path direktori untuk setiap subset data, jumlah kelas (nc), dan daftar nama kelas (names). Saat proses pelatihan dijalankan, file ini berfungsi sebagai penghubung antara dataset dan model YOLOv8. Sangat disarankan untuk memverifikasi file ini sebelum pelatihan karena salah satu sumber kesalahan yang paling sering terjadi dalam praktik penelitian adalah ketidaksesuaian informasi dalam file YAML dengan struktur direktori aktual.



Gambar 6. Dataset Berisi Folder gambar

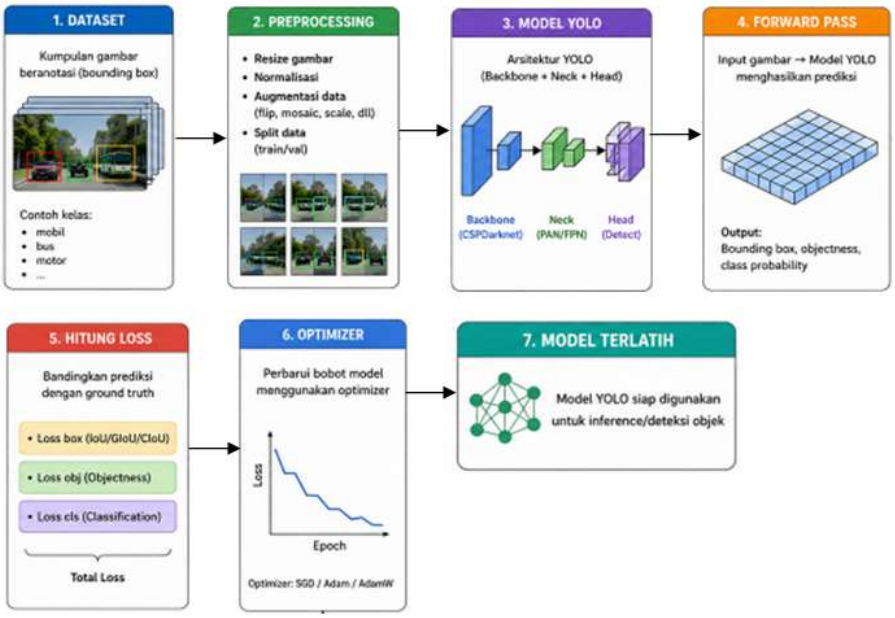


Gambar 7. Dataset Berisi Folder Label dan Text

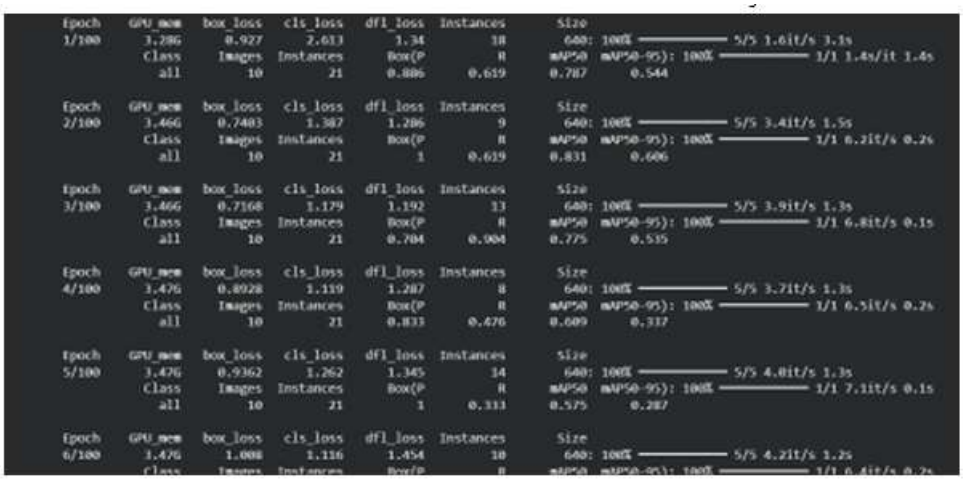
Kode Python yang ditampilkan pada hasil download dataset Roboflow berfungsi untuk menghubungkan proyek dataset di Roboflow dengan environment coding seperti Google Colab atau Jupyter Notebook. Secara keseluruhan, kode ini berfungsi sebagai jembatan otomatis antara Roboflow dan environment machine learning. Dengan menggunakan script tersebut, pengguna tidak perlu mengunduh dataset secara manual dalam bentuk ZIP karena dataset langsung diambil dari cloud Roboflow dan diubah ke format YOLOv8 sehingga siap digunakan untuk pelatihan model object detection. Proses ini mempercepat proses computer vision, mulai dari anotasi data di Roboflow hingga pelatihan model di Google Colab atau Jupyter Notebook. Gambar lain juga menunjukkan file dataset hasil export. Secara keseluruhan, struktur file menunjukkan bahwa dataset telah disiapkan secara sistematis untuk pipeline object detection berbasis YOLOv8, termasuk pembagian data untuk instruksi, validasi, pengujian, dan konfigurasi otomatis. Dengan struktur seperti ini, dataset dapat digunakan secara langsung pada lingkungan seperti Google Colab atau Jupyter Notebook untuk proses instruksi, evaluasi, dan penerapan model computer vision.

2.4 Training (Pelatihan Model YOLOv8)

Pelatihan model adalah proses di mana YOLOv8 belajar mendeteksi objek dengan menyesuaikan nilai parameter internalnya berdasarkan data pelatihan. Proses ini dilakukan berulang kali dengan menggunakan mekanisme propagasi maju untuk menghasilkan prediksi, perhitungan fungsi loss untuk mengukur kesalahan prediksi, dan propagasi balik untuk memperbarui parameter berdasarkan gradien kesalahan. Untuk ekstraksi fitur visual, YOLOv8 menggunakan arsitektur Cross Stage Partial Network (CSPNet), yang dapat menangkap fitur dari berbagai skala resolusi [1], sehingga efektif untuk mendeteksi objek dengan ukuran yang berbeda. Feature Pyramid Network (FPN) yang terintegrasi dalam YOLOv8 memungkinkan model untuk menggabungkan fitur dari berbagai lapisan [1], yang meningkatkan kemampuan deteksi objek besar dan kecil. Dalam YOLOv8, fungsi loss terdiri dari tiga komponen utama. Pertama, localization loss mengukur ketepatan prediksi koordinat bounding box menggunakan metrik CIoU (Complete Intersection over Union) [1]. Kedua, objectness los mengukur kepercayaan model terhadap keberadaan objek dalam suatu sel grid. Terakhir, classification loss mengukur ketepatan prediksi kategori objek dengan menggunakan cross entropy biner. Proses pembelajaran, ukuran batch, jumlah epoch, dan ukuran input gambar adalah parameter penting dalam pelatihan YOLOv8. Kecepatan konvergensi dan kualitas model yang dihasilkan sangat dipengaruhi oleh pemilihan nilai parameter yang tepat.



Gambar 8. Proses Model Pelatihan YOLO untuk Deteksi Objek



Gambar 9. Proses Training model YOLOv8 Pada Google Colab

Pada penelitian ini model YOLOv8 dilatih selama 100 epoch. Ini dipilih karena bahwa pada frekuensi pelatihan tersebut model telah mencapai konvergensi yang stabil, yang ditunjukkan oleh penurunan nilai loss dan peningkatan metrik evaluasi selama proses pelatihan. Pada seluruh skenario eksperimen, parameter epoch dijaga konstan. Ini dilakukan untuk memastikan bahwa perubahan dalam performa model, terutama dalam hal akurasi bounding box, sepenuhnya dipengaruhi oleh variasi kualitas data citra daripada variasi dalam durasi atau intensitas proses pelatihan. Dalam konteks ini, epoch didefinisikan sebagai satu siklus penuh di mana model memproses dataset pelatihan dari awal hingga akhir. Dengan menetapkan jumlah epoch yang sama untuk setiap percobaan, konsistensi proses pembelajaran model dapat dijaga, sehingga hasil evaluasi menjadi lebih objektif dan variasi kualitas gambar dapat dibandingkan. Tujuan lain dari penggunaan jumlah epoch yang cukup adalah untuk mengurangi kemungkinan underfitting dan overfitting, sehingga kinerja model yang dihasilkan diharapkan menunjukkan kemampuan generalisasi terbaik terhadap variasi kualitas citra yang diuji.

2.5 Validasi Model

Dalam menentukan seberapa baik model yang telah dilatih dan juga mampu mendeteksi objek pada data yang belum pernah dilihat sebelumnya (unseen data) maka validasi model dilakukan dengan menggunakan set validasi dan tes yang berbeda dari set pelatihan. Ini memberikan estimasi performa yang tidak bias. Serangkaian metrik kuantitatif dibuat selama proses validasi, yang menunjukkan kemampuan model dalam berbagai aspek. Untuk validasi model deteksi objek, metrik utama yang digunakan adalah Intersection over Union (IoU) dan Mean Average Precision (mAP). Kedua metrik ini mengukur akurasi bounding box yang diprediksi model dibandingkan dengan anotasi ground truth. Validasi juga mencakup analisis precision-recall curve untuk memahami trade-off antara presisi dan recall model pada berbagai threshold kepercayaan. Ini dilakukan selain IoU dan mAP. Analisis false positive dan false negative membantu dalam menentukan kelemahan khusus model yang perlu diperbaiki, sementara confusion matrix digunakan untuk mengidentifikasi pola kesalahan klasifikasi.



Gambar 10. Hasil Tes Deteksi Objek YOLOv8

3. HASIL DAN ANALISA

Metrik kualitas data citra adalah ukuran kuantitatif yang digunakan untuk menilai berbagai aspek kualitas gambar digital. Resolusi (px), Perbandingan Peak Signal-to-Noise (PSNR), Structural Similarity Index Measure (SSIM) dan Intersection over Union (IoU) serta Mean Average Precision (mAP@0.5) adalah lima metrik utama yang digunakan dalam penelitian ini. Untuk memberikan gambaran mendalam tentang kualitas data sepanjang proses, kelima metrik ini diukur pada berbagai tahapan pipeline.

3.1 Perhitungan Resolusi (px)

Resolusi citra adalah istilah yang mengacu pada jumlah piksel yang terdiri dari sebuah gambar digital. Format ini biasanya digambarkan dalam ukuran lebar x tinggi, misalnya 640 x 480 piksel. Karena resolusi menunjukkan seberapa detail visual yang tersedia dalam sebuah gambar, resolusi adalah metrik kualitas yang paling penting. Resolusi gambar sangat penting dalam pipeline deteksi objek berbasis YOLOv8. Resolusi asli gambar menentukan seberapa banyak informasi visual yang tersedia untuk proses anotasi pada tahap pertama persiapan data citra. Kedua, selama tahap pelatihan pada YOLOv8 mengubah ukuran semua gambar yang dimasukkan menjadi ukuran standar, yang default adalah 640 x 640 piksel [2]. Ini berarti bahwa gambar dengan resolusi lebih rendah dari standar akan diunggah, yang dapat mengurangi kualitas visual. Dua cara untuk memahami bagaimana resolusi memengaruhi akurasi bounding box adalah mekanisme pertama terkait dengan anotasi. Dengan gambar beresolusi tinggi, anotator (baik manusia maupun algoritma otomatis) dapat menentukan batas objek dengan lebih akurat, yang menghasilkan ground truth yang lebih akurat. Mekanisme kedua terkait dengan pembelajaran model adalah resolusi yang cukup untuk memastikan bahwa lapisan konvolusi dalam arsitektur YOLOv8 masih dapat mengidentifikasi fitur visual yang relevan, terutama untuk objek berukuran kecil. Dalam kebanyakan kasus, meningkatkan resolusi dari 320px ke 640px memberikan peningkatan mAP yang signifikan [2]; namun, meningkatkan resolusi dari 640px ke 1280px memberikan manfaat yang lebih sedikit dengan konsekuensi waktu komputasi yang lebih lama. Ini menunjukkan bahwa ada titik resolusi ideal yang bergantung pada karakteristik dataset dan ukuran objek yang dituju.

Resolusi Efektif (Reff)

$$R_{eff} = \min (R_{orig}, R_{model})$$

Keterangan:

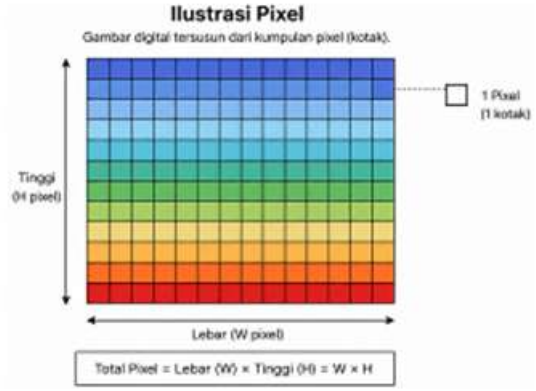
R_{eff} = Resolusi efektif
R_{orig} = Resolusi asli gambar
R_{model}= Resolusi input model YOLO

Total Piksel

$$TP = W \times H$$

Keterangan:

TP = Total piksel
W = Lebar gambar (pixel)
H = Tinggi gambar (pixel)



Gambar 11. Ilustrasi Pixel

Dalam penelitian ini digunakan sebanyak 50 citra yang memiliki variasi resolusi yang berbeda-beda. Data citra yang dianalisis merupakan dataraw yang belum melalui tahap seleksi yang akan digunakan langsung dalam proses pelatihan model, sehingga bukan merupakan dataset. Pengukuran resolusi dilakukan untuk mengetahui karakteristik dimensi setiap citra, yang dinyatakan dalam satuan piksel (pixel), yaitu lebar dan tinggi gambar. Proses perhitungan resolusi dilakukan secara otomatis menggunakan bahasa pemrograman Python guna meningkatkan efisiensi dan akurasi analisis. Dengan pendekatan ini, setiap citra dalam dataset dapat diidentifikasi resolusinya secara sistematis, sehingga memudahkan dalam proses pengelompokan dan evaluasi kualitas data citra yang digunakan dalam pelatihan model YOLOv8.

```python
import os
import cv2

# Gunakan raw string (r"...") untuk path Windows
folder_path = r"C:\Users\Gie\training\dataraw"

# Cek apakah folder ada
if not os.path.exists(folder_path):
    print(f"Folder tidak ditemukan: {folder_path}")
else:
    print("Daftar Resolusi Gambar:\n")

    for file in os.listdir(folder_path):
        if file.lower().endswith((".jpg", ".png", ".jpeg")):
            path = os.path.join(folder_path, file)

            img = cv2.imread(path)

            if img is not None:
                height, width, channels = img.shape
                print(f"{file} -> {width} x {height}")
            else:
                print(f"{file} -> gagal dibaca")
```

Gambar 12. Kode Python untuk Menghitung Resolusi (pixel)

Dari hasil perhitungan di atas didapat data dengan kriteria Berdasarkan standar input YOLO (640×640) dan konvensi computer vision:

Tabel 1. Hasil Perhitungan Resolusi (px)

Kategori	Kriteria	Dasar	Jumlah	Persentase
Tinggi	Kedua sisi ≥ 640px	Memenuhi standar input YOLO	33	66%
Sedang	Sisi terpendek 480–639px	Mendekati standar, detail cukup	12	24%
Rendah	Sisi terpendek < 480px	Di bawah standar, detail berkurang	5	10%

Tabel ini menunjukkan hasil pengelompokan 50 gambar dataset kelinci ke dalam tiga kategori kualitas berdasarkan ukuran resolusi gambar. Hanya 1 gambar (2%) yang masuk kategori ini, yaitu gambar dengan kedua sisinya mencapai 640px atau lebih. Gambar seperti ini sudah sesuai dengan ukuran input standar YOLO sehingga tidak perlu penyesuaian besar saat proses training. Sebanyak 17 gambar (34%) masuk kategori sedang, dengan sisi terpendek berada di antara 480px hingga 639px. Gambar-gambar ini masih cukup layak karena detailnya terjaga, meskipun belum sepenuhnya memenuhi standar YOLO. Saat di-resize ke 640×640, penurunan kualitas yang terjadi masih dalam batas wajar. Ini adalah kelompok terbesar, yaitu 32 gambar atau 64% dari total dataset. Sisi terpendek gambar di kelompok ini di bawah 480px, artinya resolusinya jauh dari standar input YOLO. Ketika gambar seperti ini diperbesar paksa ke 640×640, detail objek akan berkurang dan tepi objek bisa menjadi kurang tajam, yang berpotensi mempengaruhi ketepatan bounding box saat deteksi.

3.2 Perhitungan Peak Signal to Noise Ratio (PSNR)

Peak Signal-to-Noise Ratio (PSNR) adalah rasio antara kekuatan sinyal gambar asli terhadap kekuatan noise atau distorsi yang ada dalam gambar [6]. PSNR, yang dinyatakan dalam satuan desibel (dB), adalah salah satu metrik kualitas gambar yang paling banyak digunakan dalam literatur pengolahan citra digital. PPSNR dihitung dengan memanfaatkan Mean Squared Error (MSE), yang merupakan rata-rata kuadrat selisih nilai piksel antara gambar asli dan yang telah mengalami distorsi. Di sisi lain, distorsi yang besar menghasilkan MSE yang tinggi dan PSNR yang rendah, sementara nilai MSE yang rendah menunjukkan distorsi yang terjadi sangat kecil.

1. Mean Squared Error (MSE)

$$MSE = \frac{1}{mn} \sum_{i=1}^{m} \cdot \sum_{j=1}^{n} [I(i,j) - K(i,j)^2$$

Keterangan:
MSE = Mean Squared Error
m = jumlah baris piksel gambar
n = jumlah kolom piksel gambar
I (i, j) = nilai piksel citra asli pada koordinat (i, j)
K (i, j) = nilai piksel citra hasil/proses pada koordinat (i, j)

2. Peak Signal-to-Noise Ratio (PSNR)

$$PNSR = 10 \text{ Log } _{10} \left(\frac{MAX^2}{MSE} \right)$$

Keterangan:
PSNR = Peak Signal-to-Noise Ratio (dB)
MAX = nilai maksimum piksel gambar untuk citra 8-bit: MAX = 255
MSE = Mean Squared Error

Untuk membandingkan kualitas gambar sebelum dan sesudah augmentasi, PSNR diukur pada tahap dataset dalam penelitian ini. Penambahan noise Gaussian, kompresi JPEG, atau perubahan kecerahan dapat menyebabkan penurunan nilai PSNR. Pemantauan PSNR setelah peningkatan sangat penting untuk memastikan bahwa data pembelajaran tetap memiliki kualitas visual yang cukup untuk mendukung pembelajaran model yang efektif. Salah satu keterbatasan utama PSNR adalah bahwa metrik ini hanya menghitung tingkat piksel tanpa mempertimbangkan struktur spasial gambar atau persepsi visual manusia. Dua gambar dengan nilai PSNR yang sama dapat memiliki kualitas persepsi yang sangat berbeda tergantung pada distribusi spasial distorsinya. Karena keterbatasan ini, PSNR biasanya digunakan bersama dengan SSIM.

Dalam penelitian ini digunakan sebanyak 50 data citra yang akan dihitung nilai Peak Signal-to-Noise Ratio (PSNR) dengan menggunakan bahasa pemrograman Python. Perhitungan PSNR dilakukan dengan membandingkan setiap citra uji terhadap citra referensi yang dianggap memiliki kualitas lebih baik atau sebagai ground truth. Tujuan dari pengukuran ini adalah untuk mengetahui tingkat degradasi kualitas citra yang disebabkan oleh noise, kompresi, atau proses manipulasi lainnya. Nilai PSNR dinyatakan dalam satuan desibel (dB), di mana semakin tinggi nilai PSNR maka semakin baik kualitas citra tersebut karena tingkat kemiripannya terhadap citra referensi semakin tinggi. Sebaliknya, nilai PSNR yang rendah menunjukkan adanya perbedaan yang signifikan antara citra uji dan citra referensi. Hasil perhitungan PSNR ini selanjutnya digunakan sebagai salah satu parameter dalam menganalisis pengaruh kualitas citra terhadap akurasi bounding box pada model YOLOv8.

Tabel. 2 Interpretasi Nilai PSNR

Rentang PSNR (dB)	Kualitas	Interpretasi untuk Deteksi Objek	Jumlah
< 20 dB	Sangat Buruk	Gambar tidak layak untuk anotasi dan training	0
20 - 30 dB	Buruk - Sedang	Anotasi sulit, akurasi bbox rendah	16
30 - 40 dB	Baik	Kualitas acceptable untuk training YOLO	32
40 - 50 dB	Sangat Baik	Anotasi mudah, akurasi bbox optimal	2
> 50 dB	Excellent	Kualitas tinggi, performa model optimal	0

Pada rentang 20–30 dB (buruk hingga sedang) terdapat sebanyak 16 citra. Citra pada kategori ini umumnya memiliki tingkat noise atau distorsi yang cukup tinggi sehingga menyulitkan proses anotasi. Dampaknya, model berpotensi mengalami penurunan akurasi dalam menentukan bounding box karena objek tidak terdefinisi dengan jelas. Selanjutnya, pada rentang 30–40 dB (baik) terdapat 31 citra, yang merupakan jumlah terbesar dalam dataset. Kualitas citra pada rentang ini masih tergolong layak dan dapat digunakan untuk proses pelatihan model YOLOv8. Objek dalam citra masih dapat dikenali dengan cukup jelas sehingga anotasi dapat dilakukan dengan relatif baik, meskipun belum optimal. Pada rentang 40–50 dB (sangat baik) terdapat 2 citra. Citra dalam kategori ini memiliki tingkat kemiripan yang tinggi dengan citra referensi, sehingga proses anotasi menjadi lebih mudah dan akurasi bounding box yang dihasilkan model cenderung lebih optimal.

3.2 Perhitungan Structural Similarity Index Measure (SSIM)

Structural Similarity Index Measure (SSIM) digunakan untuk menilai kemiripan perseptual antara dua gambar dengan mempertimbangkan tiga elemen struktural: luminansi (kecerahan), kontras, dan struktur, SSIM menganalisis pola spasial dalam gambar, yang lebih mencerminkan cara sistem visual manusia melihat kualitas gambar, berbeda dengan PSNR yang beroperasi pada tingkat piksel tertentu

1. Rumus umum Structural Similarity Index Measure (SSIM)

$$SSIM\ (x, y) = [l(x,y)]^{\alpha} \text{ x } [c(x,y)]^{\beta} \text{ x } [s(x,y)]^{\gamma}$$

Keterangan:
$l(x,y)$= komponen luminance (kecerahan)
$c(x,y)$= komponen contrast (kontras)
$s(x,y)$= komponen structure (struktur)

2. Rumus implementasi SSIM yang umum digunakan

$$SSIM\ (x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\left(\mu_x^2+\mu_y^2+C_1\right)\left(\sigma_x^2+\sigma_y^2+C_2\right)}$$

Keterangan:
μ_x= *rata-rata intensitas piksel citra* x
μ_y= rata-rata intensitas piksel citra y
σ_x^2= varians citra x
σ_y^2= varians citra y
σ_{xy}= kovarians antara citra $x$dan y
C_1 dan C_2= konstanta stabilisasi untuk menghindari pembagian nol



Gambar 13. Interpretasi Nilai SSIM

Nilai SSIM umumnya dianggap baik dalam praktik pengolahan gambar untuk deep learning, dengan nilai di atas 0,8 dianggap baik, nilai di antara 0,6 dan 0,8 dianggap acceptable, dan nilai di bawah 0,6 menunjukkan degradasi struktural yang signifikan. Nilai SSIM dalam rentang -1 hingga 1 menunjukkan kedua gambar identik sempurna. Karena model YOLO bergantung pada atribut struktural seperti tepi, tekstur, dan pola spasial untuk mendeteksi dan melokalisasi objek, SSIM sangat relevan untuk penelitian ini. Struktur visual objek telah berubah secara signifikan jika augmentasi data menghasilkan gambar dengan SSIM yang sangat rendah dibandingkan gambar aslinya. Akibatnya, model mungkin kesulitan menemukan objek yang sama di dunia nyata. Evaluasi anotasi otomatis memerlukan SSIM juga. Model anotasi otomatis seperti SAM menemukan batas struktur dalam gambar. Gambar dengan SSIM rendah, yang disebabkan oleh noise atau distorsi, memiliki batas struktur yang ambigu, sehingga model anotasi menghasilkan bounding box yang kurang akurat. Selama proses pelatihan, bounding box yang tidak tepat ini menjadi ground truth yang tidak sempurna.

Tabel 3. Hasil Perhitungan SSIM

Kategori	Kriteria	Jumlah
Gambar Identik	1	
Kemiripan tinggi	0.75 - 1	49
Kemiripan sedang	0.50 – 0.75	1
Kemiripan rendah	0.25 – 0.50	
Sangat tidak mirim	0 – 0.25	

Berdasarkan hasil pengukuran Structural Similarity Index Measure (SSIM) terhadap 50 citra dalam dataset, dapat diketahui tingkat kemiripan struktur antara citra uji dengan citra referensi. SSIM merupakan metrik yang menilai kesamaan citra berdasarkan aspek luminansi, kontras, dan struktur, dengan rentang nilai antara 0 hingga 1, di mana nilai yang mendekati 1 menunjukkan tingkat kemiripan yang sangat tinggi. Dari tabel tersebut, terlihat bahwa sebanyak 49 citra berada pada kategori kemiripan tinggi (0.75 – 1). Hal ini menunjukkan bahwa sebagian besar citra dalam dataset memiliki kualitas yang sangat baik dan memiliki struktur visual yang hampir sama dengan citra referensi. Kondisi ini sangat mendukung proses anotasi serta pelatihan model YOLOv8, karena objek dalam citra dapat dikenali dengan jelas sehingga berpotensi menghasilkan akurasi bounding box yang tinggi. Sementara itu, terdapat 1 citra yang masuk dalam kategori kemiripan sedang (0.50 – 0.75). Citra pada kategori ini masih dapat digunakan, namun terdapat sedikit perbedaan struktur yang dapat mempengaruhi ketepatan deteksi objek, terutama pada kondisi tertentu seperti noise atau penurunan kualitas visual. Tidak ditemukan citra pada kategori kemiripan rendah (0.25 – 0.50), sangat tidak mirip (0 – 0.25), maupun gambar identik (SSIM = 1). Hal ini menunjukkan bahwa dataset tidak mengandung citra dengan kualitas sangat buruk maupun citra yang benar-benar identik dengan referensi.

3.3 Perhitungan Intersection over Union (IoU)

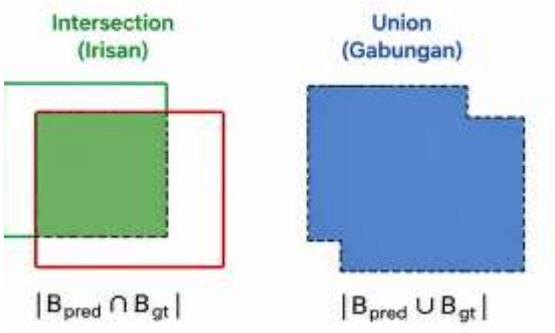
Intersection over Union (IoU) adalah metrik geometris yang mengukur tingkat tumpang tindih antara bounding box yang diprediksi oleh model dengan box ground truth yang telah dianotasi. Karena IoU memberikan ukuran yang intuitif dan mudah diinterpretasikan tentang akurasi lokasi objek, ia menjadi dasar dari hampir semua metrik evaluasi dalam deteksi objek modern [6], [13].

Rumus Intersection over Union (IoU)

$$IoU = \frac{Area\ Intersaction}{Area\ Union}$$

atau notasi matematis :

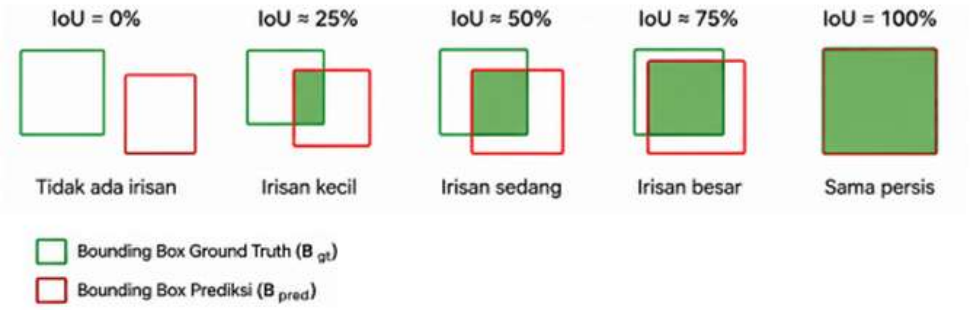
$$IoU = \frac{|B_{pred} \cap B_{gt}|}{|B_{pred} \cup B_{gt}|}$$



Gambar 14. Irisan dan Gabungan IoU

Keterangan

B_{pred} = bounding box hasil prediksi model
B_{gt} = bounding box ground truth (label aktual)
∩ = area irisan (*intersection*) antara dua bounding box
∪ = area gabungan (*union*) antara dua bounding box
|.|= luas area bounding box



Gambar 15. Contoh Nilai IoU

Nilai IoU berkisar antara 0 dan 1; nilai 0 menunjukkan bahwa tidak ada tumpang tindih sama sekali antara bounding box prediksi dan ground truth, sementara nilai 1 menunjukkan bahwa kedua kotak batas identik sempurna. Dalam praktik evaluasi model, prediksi dianggap sebagai True Positive (TP) jika nilai IoU-nya melebihi ambang tertentu, biasanya 0,5 [9], sebaliknya, jika ambang tersebut lebih rendah, prediksi dianggap sebagai False Positive (FP). Penelitian ini memiliki dua manfaat dari penggunaan Intersection over Union (IoU). Pertama, IoU digunakan sebagai metrik evaluasi model pada tahap validasi untuk mengukur akurasi bounding box yang diprediksi YOLOv8. Kedua, IoU antara anotasi yang dibuat secara otomatis dan anotasi manual referensi dapat menunjukkan seberapa akurat sistem anotasi otomatis bekerja pada gambar dengan berbagai tingkat kualifikasi. Beberapa jenis IoU telah dikembangkan untuk mengatasi keterbatasan IoU konvensional. GIoU (Generalized IoU) menambahkan penalti untuk kasus di mana kedua bounding box tidak bersinggungan [1]. Sebaliknya, DIoU (Distance IoU) mempertimbangkan jarak antara pusat kedua bounding box. Dalam fungsi loss YOLOv8, CIoU (Complete IoU) secara bersamaan mempertimbangkan overlap aspek, jarak pusat, dan rasio aspek. Dengan demikian, konvergensi pelatihan lebih cepat dan stabil.

Evaluasi model deteksi objek dilakukan menggunakan framework Ultralytics YOLOv8 dengan memanfaatkan bobot model terbaik (best.pt) yang diperoleh dari proses pelatihan Proses evaluasi dilakukan dalam beberapa tahap antara lain yang pertama, model YOLOv8 yang telah dilatih digunakan untuk melakukan inferensi terhadap seluruh citra pada dataset validasi Kedua, nilai IoU dihitung secara manual untuk setiap pasangan bounding box prediksi dan ground truth. Format koordinat YOLO terlebih dahulu dikonversi ke format corner-based. Untuk setiap bounding box ground truth, dipilih nilai IoU tertinggi (best match) dari seluruh bounding box hasil prediksi pada citra yang sama. Apabila tidak terdapat hasil prediksi pada suatu citra, maka nilai IoU untuk bounding box ground truth pada citra tersebut ditetapkan sebagai 0. Ketiga, seluruh nilai IoU yang diperoleh direkap dan dikelompokkan ke dalam lima rentang nilai sebagaimana ditunjukkan pada Tabel berikut.

Tabel 4. Interpretasi Nilai IoU:

Rentang IoU	Kategori	Interpretasi	Jumlah Box
0.0 - 0.3	Sangat Rendah	Prediksi sangat meleset dari objek	2
0.3 - 0.5	Rendah	Lokalisasi tidak akurat, deteksi gagal	2
0.5 - 0.7	Sedang	Deteksi berhasil (standar minimal PASCAL VOC)	2
0.7 - 0.9	Tinggi	Lokalisasi akurat, kualitas baik	14
0.9 - 1.0	Sangat Tinggi	Lokalisasi hampir sempurna	88
Total box dari 50 data Citra			108
Mean IoU keseluruhan			0.9156

Proses rekapitulasi dilakukan dengan menghitung jumlah bounding box (box) yang nilai IoU-nya jatuh pada masing-masing rentang tersebut. Satuan perhitungan yang digunakan adalah bounding box dan bukan citra, karena nilai IoU secara inheren merupakan ukuran akurasi pada level bounding box, di mana satu citra dapat mengandung lebih dari satu objek dengan nilai IoU yang berbeda-beda.

Keseluruhan proses evaluasi diimplementasikan menggunakan bahasa pemrograman Python dengan memanfaatkan library Ultralytics YOLOv8 untuk proses inferensi, NumPy untuk komputasi numerik. Hasil evaluasi menunjukkan bahwa dari total 108 bounding box yang terdapat pada 50 citra validasi, sebanyak 88 box (81,5%) memperoleh nilai IoU pada rentang 0,9–1,0 yang termasuk kategori sangat baik, dan 14 box (13,0%) berada pada rentang 0,7–0,9 kategori baik. Hal ini mengindikasikan bahwa model YOLOv8 yang dilatih mampu melakukan deteksi objek dengan tingkat akurasi lokalisasi yang tinggi pada dataset validasi yang digunakan.

3.4 Perhitungan Mean Average Precision at IoU 0.5 (mAP@0.5)

Mean Average Precision at IoU threshold 0.5 (mAP@0.5) adalah metrik evaluasi menyeluruh yang mengukur performa model deteksi objek secara keseluruhan dengan mempertimbangkan akurasi lokalisasi

dan akurasi klasifikasi objek. mAP@0.5 adalah standar yang paling umum digunakan di industri dan lingkungan akademik untuk membandingkan kinerja model deteksi objek yang berbeda. Perhitungan mAP@0.5 dilakukan melalui berbagai langkah yang saling berhubungan. Pertama, model menghasilkan prediksi bounding box dan skor kepercayaan (confidence score) untuk setiap kelas objek. Prediksi disusun berdasarkan skor kepercayaan, diurutkan dari yang tertinggi ke yang terendah. Setelah itu, setiap prediksi dievaluasi menggunakan threshold (IoU) 0,5. Prediksi dengan ambang intelegensi lebih dari 0,5 dianggap sebagai prediksi True Positivei (TP), sedangkan prediksi dengan ambang intelegensi yang kurang dari 0,5 dianggap sebagai prediksi False Positive (FP).

1. Precision

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

Keterangan

TP = True Positive, objek terdeteksi benar
FP = False Positive, model mendeteksi objek yang sebenarnya salah/tidak ada

2. Recall

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

Keterangan

TP = True Positive, objek terdeteksi benar
FN = False Negative, objek sebenarnya ada tetapi gagal terdeteksik

3. Average Precision (AP)

$$AP = \int_0^1 p(r)dr$$

Dalam implementasi diskrit biasanya dihitung sebagai:

$$AP \approx \sum_n (R_n - R_{n-1}) P_n$$

Keterangan

$p(r)$ = precision sebagai fungsi recal
R_n= recall pada threshold ke-n
R_{n-1}= recall sebelumnya

4. Mean Average Precision at IoU 0.5 (mAP@0.5)

$$mAP\ @0.5 = \frac{1}{N}\sum_{i=1}^{N} AP_i$$

Keterangan

N = jumlah kelas objek
AP_i = Average Precision kelas ke-i pada threshold IoU 0.5

mAP@0.5 adalah rata-rata AP seluruh kelas dengan kriteria prediksi benar jika:

IoU≥0.5

Average Precision (AP) untuk setiap kelas adalah luas area yang berada di bawah kurva Recall Precision (PR). Kurva ini dibuat dengan mengubah nilai threshold confidence dan menghitung pasangan nilai precision dan recall yang sesuai. Nilai AP yang tinggi menunjukkan bahwa model memiliki precision yang tinggi pada semua tingkat precision , yang berarti bahwa model dapat mendeteksi hampir semua objek yang ada sambil tetap mempertahankan tingkat precision yang rendah. Nilai AP untuk setiap kelas objek dirata-ratakan untuk mendapatkan mAP@0.5. Dengan menggunakan ambang IoU threshold 0,5, sebuah prediksi dianggap benar hanya jika bounding box yang diprediksi memiliki overlap dengan ground truth minimal 50%.

Evaluasi performa model deteksi objek berbasis YOLOv8 dilakukan menggunakan fungsi dari framework Ultralytics yang menjalankan ulang proses validasi secara langsung terhadap dataset validasi. Metode ini dipilih karena perhitungan metrik dilakukan secara real time dengan inferensi penuh terhadap seluruh citra validasi, sehingga menghasilkan nilai yang paling akurat dan dapat dipertanggungjawabkan secara ilmiah dan berikut hasilnya disampaikan pada tabel di bawah ini :

Tabel 5. Interpretasi Nilai mAP@0.5:

Rentang mAP@0.5	Rentang mAP@0.5:0.95	Kualitas Model	Rekomendasi	Hasil mAP@0.5	Hasil mAP@0.5:0.95
< 0.30	< 0.20	Sangat Rendah	Model tidak layak, perlu perbaikan dataset	-	-
0.30 - 0.50	0.20 – 0.40	Rendah	Perlu perbaikan signifikan pada data dan arsitektur	-	-
0.50 - 0.70	0.40 – 0.60	Sedang	Acceptable untuk aplikasi non-kritis	-	-
0.70 - 0.85	0.60 – 0.75	Baik	Layak untuk deployment dengan monitoring	-	-
> 0.85	> 0.75	Sangat Baik	Performa tinggi, siap untuk produksi	0.9836	0.8978

Proses evaluasi menggunakan bobot model terbaik (best.pt) yang diperoleh dari proses pelatihan, dengan nilai threshold IoU ditetapkan sebesar 0,5 sesuai standar evaluasi COCO (Common Objects in Context) [1]. Hasil evaluasi menunjukkan bahwa model YOLOv8 yang dilatih memperoleh nilai mAP@0.5 sebesar 0,9836 atau 98,36%. Berdasarkan kriteria kualitas model yang digunakan, nilai tersebut berada pada rentang > 0,85 yang dikategorikan sebagai "Sangat Baik", mengindikasikan bahwa model memiliki performa tinggi dan siap untuk digunakan pada tahap produksi. Selain itu, model juga dievaluasi menggunakan metrik mAP@0.5:0.95 yang memperoleh nilai sebesar 0,8978 atau 89,78%. Metrik ini merupakan rata-rata nilai mAP pada rentang threshold IoU dari 0,5 hingga 0,95 dengan interval 0,05, sehingga memberikan gambaran yang lebih komprehensif terhadap kemampuan lokalisasi objek pada berbagai tingkat ketetatan (strictness) deteksi. Nilai mAP@0.5:0.95 sebesar 89,78% mengkonfirmasi bahwa model tidak hanya mampu mendeteksi objek dengan akurat pada threshold standar, tetapi juga mempertahankan performa yang tinggi pada threshold yang lebih ketat.

Secara keseluruhan, hasil evaluasi ini menunjukkan bahwa model YOLOv8 yang dikembangkan memiliki kemampuan deteksi objek yang sangat baik, dengan tingkat presisi dan akurasi lokalisasi yang tinggi pada dataset validasi yang digunakan. Hal ini juga didukung oleh hasil perhitungan IoU per bounding box yang menunjukkan bahwa 88 dari 108 bounding box (81,5%) berada pada rentang IoU 0,9–1,0, yang semakin memperkuat kesimpulan bahwa model mampu memprediksi lokasi objek dengan sangat presisi

4. KESIMPULAN

Berdasarkan hasil penelitian ini dapat disimpulkan bahwa kualitas data citra memiliki pengaruh yang sangat signifikan terhadap performa deteksi objek dan akurasi bounding box pada model YOLOv8. Hasil analisis menunjukkan bahwa sebagian besar dataset memiliki kualitas citra yang baik, ditinjau dari parameter resolusi, PSNR, dan SSIM, sehingga mampu mendukung proses training dan deteksi objek secara optimal. Pada parameter resolusi, sebanyak 66% data citra termasuk kategori tinggi dengan ukuran memenuhi standar input YOLO, sehingga detail objek pada gambar dapat dipertahankan dengan baik. Hal ini membantu model dalam mengenali fitur objek secara lebih akurat. Dari sisi kualitas sinyal citra, hasil PSNR menunjukkan mayoritas data berada pada rentang 30-40 dB yang termasuk kategori baik, sedangkan sebagian kecil berada pada kategori sangat baik. Kondisi ini menunjukkan bahwa tingkat noise pada citra relatif rendah dan masih layak digunakan untuk proses anotasi maupun pelatihan model.

Analisis SSIM menunjukkan bahwa 49 dari 50 citra memiliki tingkat kemiripan struktur visual yang tinggi. Hasil ini menandakan bahwa kualitas visual dataset cukup konsisten dan stabil, sehingga model dapat mempelajari pola objek dengan lebih efektif tanpa mengalami banyak gangguan perubahan struktur citra.

Tabel 6. Kesimpulan

No	Parameter Evaluasi	Hasil	Kategori	Kesimpulan
1	Resolusi Citra	66% citra ≥ 640px	Tinggi	Mayoritas citra memenuhi standar input YOLOv8
2	PSNR	64% citra pada 30–40 dB	Baik	Kualitas citra acceptable untuk pelatihan YOLOv8
3	SSIM	98% citra pada 0,75–1,0	Kemiripan Tinggi	Konsistensi struktural dataset sangat baik
4	Mean IoU	0,9156	Sangat Tinggi	Lokalisasi bounding box hampir sempurna
5	mAP@0.5	0,9836 (98,36%)	Sangat Baik	Model siap untuk produksi
6	mAP@0.5:0.95	0,8978 (89,78%	Sangat Baik	Performa tinggi pada berbagai threshold IoU

Dari hasil evaluasi deteksi objek, diperoleh nilai Mean IoU sebesar:

Mean IoU=0.9156

Nilai tersebut termasuk kategori sangat tinggi dan menunjukkan bahwa bounding box hasil prediksi model memiliki tingkat kesesuaian yang sangat dekat dengan ground truth. Sebagian besar hasil deteksi juga berada pada rentang IoU 0.9–1.0, yang berarti lokalisasi objek hampir sempurna. Selain itu, performa model berdasarkan parameter Mean Average Precision menunjukkan hasil yang sangat baik, yaitu:

mAP@0.5 = 0.9836

dan

mAP@0.5:0.95 = 0.8978

Nilai tersebut menunjukkan bahwa model tidak hanya mampu mendeteksi objek dengan tingkat keberhasilan yang sangat tinggi, tetapi juga mampu menghasilkan bounding box yang presisi pada berbagai threshold IoU. Berdasarkan rentang interpretasi mAP, model termasuk kategori sangat baik dan layak digunakan untuk implementasi sistem deteksi objek pada lingkungan produksi maupun aplikasi real-time. Secara keseluruhan, penelitian ini membuktikan bahwa kualitas data citra yang baik, terutama dari aspek resolusi, noise, dan kemiripan struktur visual, berkontribusi besar terhadap peningkatan akurasi deteksi objek dan kualitas bounding box pada model YOLOv8. Semakin baik kualitas citra yang digunakan, maka semakin tinggi pula performa model dalam melakukan lokalisasi objek secara akurat dan stabil.

Secara keseluruhan, hasil penelitian menunjukkan bahwa kualitas data citra yang baik, tercermin dari dominasi resolusi tinggi, nilai PSNR pada kategori baik, serta konsistensi struktural yang tinggi berdasarkan SSIM, memberikan dampak positif yang signifikan terhadap akurasi bounding box model YOLOv8. Hal ini dibuktikan dengan nilai Mean IoU sebesar 0,9156 dan mAP@0.5 sebesar 98,36% yang keduanya berada pada kategori sangat baik. Dengan demikian, dapat disimpulkan bahwa kualitas data citra memiliki peran yang krusial dalam menentukan akurasi dan performa model deteksi objek berbasis YOLOv8.

Referensi

[1] Shital Katkade, Dr. Ramesh R. Manza. Performance Evaluation of YOLOv8 for Object Detection on the COCO Dataset. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*.2025; 13(9):1531-1538

[2] Vivi Afifah, Surni Erniwati. YOLOv8 for Object Detection: A Comprehensive Review of Advances, Techniques, and Applications. *International Jurnal of Advanced Computing and Informatics.* 2026; 2(1): 53-61.
[3] Minal A. Zope, Rakesh K. Deshmukh, Priya Pise. Enhanced Alzheimer's Disease Abnormality Classification in Medical Imaging Using YOLOv8. *SSRG International Journal of Computer Science and Engineering*. 2025; 12(2):19-28.
[4] Erinna Astiadewi, Martanto, Arif Rinaldi Dikananda. Algoritma YOLOv8 untuk Meningkatkan Analisa Gambar Dalam Mendeteksi Jerawat. *Jurnal Informatika Teknologi dan Sains (JINTEKS)*. 2025; 7(1): 346-353.
[5] Rizky Hidayatullah, Nana Suarna, Irfan Ali. Deep Learning Algoritma YOLOv8 untuk Meningkatkan Analisis Kepadatan Lalu Lintas. *JITET (Jurnal Informatika dan Teknik Elektro Terapan)*. 2025; 13(2): 1-8.
[6] Gonzalez RC, Woods RE. Digital Image Processing. Fourth Edition. New York: Pearson Education. 2018: 25-48.
[7] Szeliski R. Computer Vision: Algorithms and Applications. Second Edition. London: Springer. 2022: 221-26.
[8] Goodfellow I, Bengio Y, Courville A. Deep Learning. Cambridge: MIT Press. 2016: 321-356.
[9] Prince SJD. Computer Vision: Models, Learning, and Inference. Cambridge: Cambridge University Press. 2012: 401-430.
[10] Russakovsky O, Fei-Fei L, Li K. Computer Vision and Pattern Recognition: A Practical Introduction. New York: Wiley. 2021: 188-214.
[11] Redmon J, Farhadi A. YOLO Object Detection and Real-Time Computer Vision. California: AI Publications. 2020: 55-78.
[12] Hartley R, Zisserman A. Multiple View Geometry in Computer Vision. Second Edition. Cambridge: Cambridge University Press. 2004: 92-110.
[13] Davies ER. Computer Vision: Principles, Algorithms, Applications, Learning. Fifth Edition. London: Academic Press. 2018: 503-548.
[14] Bradski G, Kaehler A. Learning OpenCV: Computer Vision with the OpenCV Library. Second Edition. Sebastopol: O'Reilly Media. 2016: 233-260.
[15] Bishop CM. Pattern Recognition and Machine Learning. New York: Springer. 2006: 649-683.