PENGOLAHAN CITRA · MATERI 11

YOLO Object Detection pada Gambar

Pelajari konsep melalui contoh yang dekat dengan keseharian, coba ilustrasi dan foto sendiri, lalu jalankan praktikum secara bertahap di Google Colab.

Pengantar → konsep → ilustrasi dan upload foto → video → praktikum

1. Mulai dari gambaran sederhana

Bayangkan kita ingin mengetahui banyak kendaraan pada sebuah foto jalan. Cara manualnya adalah melihat foto, menentukan benda yang termasuk kendaraan, lalu menghitungnya. Program juga membutuhkan urutan keputusan seperti itu. Ia tidak cukup diminta menghitung semua kotak, karena sebagian kotak mungkin merupakan orang atau benda lain yang bukan sasaran. Sebelum menghitung, kita perlu menyepakati kategori yang dimaksud dan memahami asal kotak deteksinya.

Bab ini melanjutkan deteksi YOLO dengan mengubah keluaran menjadi data yang dapat dibaca dan diperiksa. Kita memilih kelas target, mengambil koordinat dan confidence, lalu menghitung deteksi setiap kelas pada setiap foto. Contoh kode memakai car, motorcycle, bus, dan truck. Nama-nama itu diperiksa terhadap daftar kelas model; memilihnya tidak berarti melatih model untuk mengenali kategori baru. Jika tujuan berubah, daftar sasaran harus tetap mengikuti kategori yang tersedia.

Hal pentingnya adalah membedakan jumlah deteksi dari jumlah objek yang benar-benar terlihat. Kendaraan kecil dapat terlewat, sedangkan pola latar mungkin keliru mendapat kotak. Jumlah yang tampak cocok pun belum membuktikan semua kotak tepat, sebab kesalahan tambahan dan kehilangan objek dapat saling menutupi. Karena itu, kita membandingkan hasil otomatis dengan hitungan manual dan tetap memeriksa lokasi kotak satu per satu.

Pembahasan bergerak dari definisi sasaran, pembacaan tabel, hingga penafsiran galat hitungan. Ilustrasi dan video membantu menghubungkan foto dengan informasi detektor. Di Colab, keluaran akhirnya berupa gambar anotasi, tabel kotak, tabel hitungan, dan audit sederhana. Semua kesimpulan dibatasi pada foto yang diamati: menjumlahkan kendaraan dari beberapa foto yang menampilkan tempat sama belum berarti memperoleh jumlah kendaraan unik.

2. Pahami konsep secara berurutan

2.1 Tentukan apa yang dimaksud dengan objek target

Counting atau penghitungan objek dibangun di atas keputusan deteksi. Kita perlu menentukan kategori yang dihitung sebelum menjumlahkan kotak. Pada contoh kendaraan, `TARGET_NAMES` berisi `car`, `motorcycle`, `bus`, dan `truck`. Pengelompokan ini adalah pilihan program, bukan definisi bahwa semua alat transportasi pasti tercakup. Sepeda atau kategori lain yang tidak dipilih berada di luar sasaran hitungan tersebut.

Ketika membaca hasil, satu bounding box target yang lolos penyaringan dihitung sebagai satu deteksi. Kotak itu adalah dugaan model, bukan bukti keberadaan objek yang sudah diverifikasi. Objek kecil, kabur, atau tertutup dapat tidak mendapat kotak. Sebaliknya, sesuatu yang bukan kendaraan dapat keliru diprediksi sebagai kendaraan. Karena itu, gunakan istilah jumlah deteksi target sampai hasilnya diperiksa terhadap foto dan rujukan manusia.

2.2 Pilih kelas dari pemetaan model, bukan hafalan

`model.names` memetakan ID angka menjadi nama kategori. Kode membalik pemetaan itu menjadi `name_to_id`, sehingga pilihan dapat ditulis dalam nama yang mudah dibaca. Nama yang tidak tersedia dan daftar target kosong ditolak sebelum inferensi. Pemeriksaan ini mencegah program terlihat berjalan dengan sasaran yang sebenarnya tidak didukung model. Nama kelas harus mengikuti ejaan yang tercetak pada model yang dimuat.

Daftar `TARGET_IDS` kemudian dikirim melalui parameter `classes` untuk menyaring keluaran. Filter tidak menambah kategori dan tidak melatih ulang bobot. Bila suatu kebutuhan membutuhkan kelas baru, mengganti nama atau menebak ID tidak menyelesaikannya. Pada model custom, nama serta urutannya mengikuti data pelatihan model tersebut. Karena itu, pemetaan perlu diperiksa kembali ketika bobot diganti, walaupun kode penghitungnya tampak sama.

2.3 Ubah keluaran detektor menjadi tabel yang jelas

Setiap kotak pada `result.boxes` menyediakan kelas melalui `cls`, skor melalui `conf`, dan koordinat melalui `xyxy`. Kode mengambil nilai tersebut untuk membuat satu baris per deteksi. Pemanggilan `cpu().tolist()` memindahkan koordinat ke CPU dan mengubahnya menjadi daftar Python agar mudah disusun dalam tabel. Tabel menyimpan nama gambar, ID kelas, nama kelas, confidence, dan posisi kotak.

Kolom `width_px` dan `height_px` berasal dari selisih koordinat sudut. Keduanya adalah ukuran pada gambar, bukan ukuran fisik kendaraan. Perspektif membuat kendaraan yang sama tampak berbeda ukurannya menurut jarak kamera. Struktur kolom `detections` tetap ditetapkan walaupun tidak ada deteksi. Dengan demikian, tabel kosong masih mempunyai arti

yang jelas: tidak ada kotak yang dikembalikan pada pengaturan tersebut, bukan data yang boleh diisi dengan perkiraan.

2.4 Ringkas per kelas tanpa kehilangan konteks gambar

`Counter` menambah hitungan setiap kali sebuah kotak target dibaca. Tabel `count_table` kemudian menyimpan hitungan tiap kategori dan `total_deteksi_target` untuk setiap gambar. Ringkasan memudahkan pembacaan, sedangkan tabel kotak memungkinkan penelusuran asal angka tersebut. Keduanya saling melengkapi: ringkasan saja tidak menunjukkan kotak mana yang salah lokasi atau salah kelas.

Confidence yang dipilih melalui `CONF` memengaruhi kandidat yang dihitung. Menaikkan threshold dapat membuang prediksi keliru sekaligus objek benar yang skornya lemah. Menurunkannya dapat menambah keduanya. Karena itu, galat hitungan tidak selalu membaik hanya dengan menaikkan threshold. Jika ingin mengamati pengaruh parameter ini, gunakan foto dan filter kelas yang sama. Mengubah beberapa hal sekaligus membuat penyebab perbedaan hasil lebih sulit dijelaskan.

2.5 Buat hitungan manual yang konsisten sebelum menilai galat

Hitungan manual menjadi rujukan sederhana untuk membandingkan hasil otomatis, tetapi bukan angka yang otomatis bebas kesalahan. Tentukan kebijakan untuk kendaraan terpotong, tertutup, muncul sebagai pantulan, atau hanya tergambar pada poster. Gunakan kebijakan yang sama pada semua foto dan pada semua percobaan threshold. Jika satu foto dinilai hanya dari objek utuh sedangkan foto lain memasukkan semua potongan, perbandingan menjadi tidak seimbang.

Sel manual menampilkan gambar asli dan meminta jumlah total objek target sebagai bilangan bulat nonnegatif. Angka ini tidak disiapkan sebagai ground truth buatan. Pengamat harus menghitung sesuai foto dan definisi target, bukan menyalin jumlah deteksi model. Walaupun prompt muncul setelah inferensi pada urutan notebook, penilaian tetap perlu didasarkan pada gambar asli. Rujukan hitungan ini juga bukan anotasi kotak lengkap, sehingga belum cukup untuk mengukur ketepatan lokasi setiap deteksi secara otomatis.

2.6 Baca galat sambil tetap mengaudit kotak

Kolom `galat` dihitung dari jumlah deteksi target dikurangi jumlah manual. Tanda positif berarti hitungan otomatis berlebih terhadap rujukan; tanda negatif berarti lebih sedikit. `galat_absolut` mengabaikan arah selisih, dan rata-ratanya merangkum besar penyimpangan pada kumpulan foto yang dipakai. Ringkasan ini bukan persentase akurasi deteksi dan tidak menunjukkan objek mana yang terlewat.

Galat nol belum membuktikan semua kotak benar. Secara konseptual, sebuah objek yang terlewat dapat diimbangi oleh satu kotak palsu sehingga total tetap cocok. Pemeriksaan visual diperlukan untuk membedakan false negative, yaitu objek target yang tidak terdeteksi, dari false positive, yaitu deteksi yang tidak semestinya. Kesimpulan juga berlaku per gambar, bukan jumlah objek unik lintas foto. Kendaraan yang sama dapat muncul berulang pada foto berbeda. Dokumentasikan kondisi yang sulit, gunakan media berizin, dan batasi penggunaan angka pada cakupan pengamatan yang benar-benar dilakukan.



Count adalah deteksi pada satu gambar; bukan jumlah unik sepanjang video.

Gambar di atas dipertahankan dari materi sumber. Ilustrasi sumber tidak dengan sendirinya membuktikan kinerja pada foto lain; perhatikan keterangan, metode, dan batasannya.

---

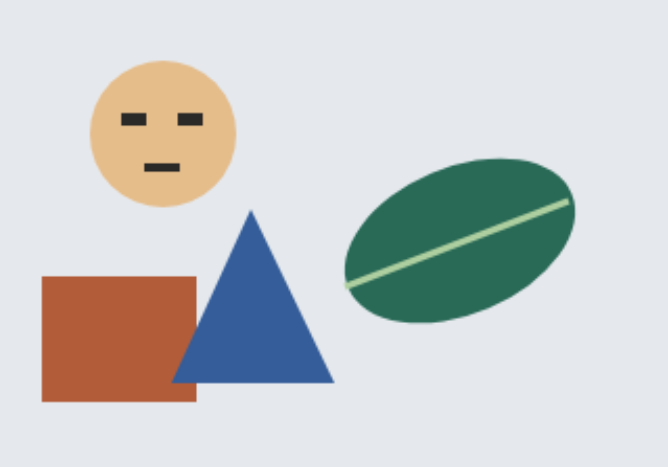
3. Ilustrasi interaktif sebelum praktikum

Ilustrasi menghitung kandidat anotasi terpilih

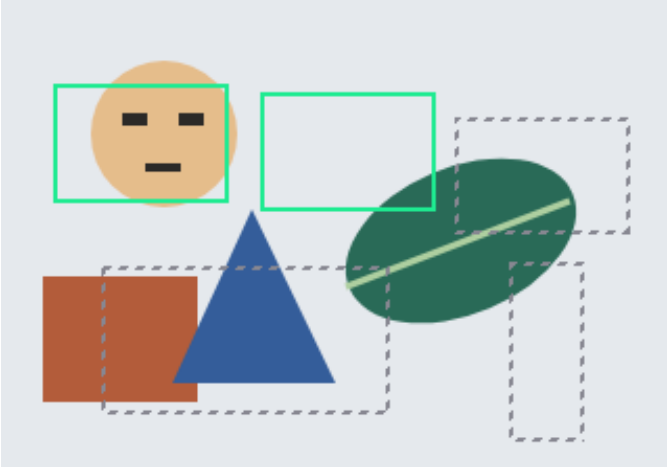
Dua preset berisi anotasi manual buatan. Kelas, posisi, dan skor tidak dibaca dari foto. Filter sungguhan menghitung kotak kandidat yang lolos, bukan objek nyata atau individu unik.

> Kandidat dan kelas pada ilustrasi adalah anotasi contoh, bukan keluaran model pada foto Anda. Jumlah yang ditampilkan adalah jumlah kandidat contoh yang lolos filter, bukan hitungan objek nyata dalam foto.

Pakai gambar contoh atau pilih foto sendiri dari HP/komputer. Foto dibaca dan diproses **hanya di browser**, tanpa dikirim atau disimpan ke server. Contoh buatan tidak digunakan sebagai bukti hasil eksperimen. Gunakan media berizin dan hindari data sensitif.



Gambar contoh sintetis · citra kerja 320 × 224 piksel



Hasil ilustrasi · baca jenis operasi dan batasannya

Parameter: Preset anotasi manual (bukan adegan terdeteksi) = A · Kelas ilustratif terpilih = car · Ambang skor buatan (tidak berasal dari foto) = 0.5

Ilustrasi manual A: 2/5 kotak kandidat lolos filter car, skor ≥ 0.50; bukan jumlah objek nyata pada foto.

Kotak hijau = kandidat lolos; abu-abu putus-putus = tersaring. Semua anotasi, nama kelas, dan skor dibuat untuk pengajaran, tidak berasal dari foto unggahan. Tidak menjalankan YOLO atau mengenali gambar. Hitungan kelas: car=2, bus=0, person=0. Tabel kandidat manual: A1: car, skor buatan 0.90, xyxy=[25.6, 40.3, 108.8, 96.3], lolos; A2: car, skor buatan 0.60, xyxy=[124.8, 44.8, 208.0, 100.8], lolos; A3: car, skor buatan 0.30, xyxy=[217.6, 56.0, 300.8, 112.0], tersaring; A4: bus, skor buatan 0.80, xyxy=[48.0, 127.7, 185.6, 197.1], tersaring; A5: person, skor buatan 0.55, xyxy=[243.2, 125.4, 278.4, 210.6], tersaring. Preset B sengaja memuat kotak berdekatan/tumpang tindih: tidak ada NMS atau deduplikasi, sehingga kotak bukan individu unik. Tidak ada ground truth atau galat empiris buatan; praktikum audit harus memakai hasil model nyata dan hitungan manual gambar asli.

▼ **Angka dan parameter hasil perhitungan ilustrasi**


```
{
  "illustration": true,
  "inference": false,
  "annotationSource": "manual-preset",
```

```
  "scene": "A",
  "targetClass": "car",
  "threshold": 0.5,
  "candidateCount": 5,
  "retainedCandidateCount": 2,
  "counts": {
    "car": 2,
    "bus": 0,
    "person": 0
  },
  "annotations": [
    {
      "id": "A1",
      "className": "car",
      "score": 0.9,
      "scoreSource": "manual",
      "xyxy": [
        25.6,
        40.32,
        108.80000000000001,
        96.32
      ],
      "retained": true
    },
    {
      "id": "A2",
      "className": "car",
      "score": 0.6,
      "scoreSource": "manual",
      "xyxy": [
        124.80000000000001,
        44.800000000000004,
        208,
        100.8
      ],
      "retained": true
    },
    {
      "id": "A3",
      "className": "car",
      "score": 0.3,
      "scoreSource": "manual",
```

```
      "xyxy": [
        217.60000000000002,
        56,
        300.79999999999995,
        112
      ],
      "retained": false
    },
    {
      "id": "A4",
      "className": "bus",
      "score": 0.8,
      "scoreSource": "manual",
      "xyxy": [
        48,
        127.67999999999999,
        185.6,
        197.12
      ],
      "retained": false
    },
    {
      "id": "A5",
      "className": "person",
      "score": 0.55,
      "scoreSource": "manual",
      "xyxy": [
        243.2,
        125.44000000000001,
        278.4,
        210.56
      ],
      "retained": false
    }
  ]
}
```


Pada PDF/cetakan, ilustrasi ini menjadi cuplikan statis menggunakan gambar contoh. Upload foto dan pengubahan parameter tersedia di versi web; foto pribadi tidak dimasukkan ke PDF unduhan.

4. Video pembelajaran

Object Detection with Pre-trained Ultralytics YOLOv8 Model | Episode 1 — Ultralytics.

Fokus saat menonton: Ikuti inferensi pretrained YOLOv8 pada media, lalu hubungkan results.boxes dengan filter kelas dan hitungan per gambar.

Buka video di YouTube jika pemutar tidak tersedia

Video referensi berbahasa Inggris. Bila ada setup lokal, API atau versi model berbeda, pelajari konsepnya dan tetap gunakan alur Google Colab pada praktikum bab ini. Ketersediaan pemutar mengikuti YouTube, pemilik video dan jaringan.

▼ **Referensi dan atribusi gambar**

- Ultralytics — Predict, Results dan Boxes
- NumPy — Dokumentasi
- Google Colab — FAQ dan keterbatasan runtime

Foto daun: Ανώνυμος Βικιπαιδιστής, Wikimedia Commons, CC BY 3.0; diubah ukuran/intensitas untuk ilustrasi. Foto bus: contoh aset Ultralytics. Foto astronaut: NASA melalui scikit-image. Panel diagram/simulasi ditandai pada caption; bukan hasil eksperimen yang dikarang. Hak penggunaan sumber tetap mengikuti sumber asli.

---

5. Praktikum / latihan Google Colab

Buka Google Colab, buat notebook baru, dan jalankan satu sel kode untuk setiap langkah secara berurutan dengan `Shift` + `Enter`. Variabel dan upload disiapkan di modul ini; tidak mengambil variabel dari bab lain.

Jalankan instalasi, pemuatan `yolov8n.pt`, dan unggah gambar secara berurutan. Sel menerima 1–20 JPG/PNG, maksimal 20 MB per gambar dan 20 megapiksel. Setelah `model.names` tercetak, periksa `TARGET_NAMES` dan `CONF`. Kode membentuk `name_to_id`, menolak nama yang tidak tersedia, lalu menyimpan filter dalam `TARGET_IDS`. Masukan tahap ini adalah foto dan definisi sasaran; belum ada hitungan manual atau penambahan kategori baru.

Jalankan sel deteksi untuk mengubah kotak aktual menjadi `detections` dan `count_table`. Cocokkan setiap baris kelas, confidence, dan koordinat dengan gambar anotasinya. Berkas `bbox.csv` dan `counting.csv` disimpan di `/content/hasil_yolo`, bersama gambar berakhiran `_count.jpg`. Hasil nol tetap sah dan tidak perlu diganti dengan kotak buatan.

Untuk mencoba threshold lain, ubah `CONF`, jalankan kembali sel pemilih kelas dan sel deteksi, lalu bandingkan keluaran pada foto yang sama; berkas dengan nama sama dapat tertimpa.

Sel berikutnya menampilkan gambar asli dan meminta jumlah manual seluruh kategori target untuk setiap gambar. Gunakan kebijakan hitung yang konsisten dan jangan menyesuaikan angka agar cocok dengan model. Variabel `audit` menggabungkan hitungan, menghitung `galat` serta `galat_absolut`, lalu menyimpan `audit_counting.csv`. Setelah setiap perubahan threshold, jalankan kembali tahap audit agar tabel yang dibandingkan sesuai dengan deteksi terbaru. Baca selisih bersama kotaknya: jumlah yang cocok tidak selalu berarti objek yang terdeteksi juga benar.

Sel 1 — Instalasi Ultralytics

Jalankan sebagai sel tersendiri di Colab; tidak perlu meng-upgrade NumPy. Catat versi paket di sel berikutnya.

```
%pip install -q ultralytics
```

Sel 2 — Model baseline dan direktori keluaran

Bobot yolov8n.pt diunduh saat pemuatan pertama. Ini baseline ringan, bukan pernyataan bahwa model ini yang terbaru. CPU dapat digunakan.

```
from pathlib import Path
import cv2, numpy as np, matplotlib.pyplot as plt
import torch, ultralytics
from ultralytics import YOLO
from google.colab import files
OUTPUT = Path('/content/hasil_yolo')
OUTPUT.mkdir(parents=True, exist_ok=True)
DEVICE = 0 if torch.cuda.is_available() else 'cpu'
model = YOLO('yolov8n.pt')
print('Ultralytics:', ultralytics.__version__, 'Device:', DEVICE)
print('Kelas model:', model.names)
```

Sel 3 — Upload dan validasi gambar

Pilih satu atau lebih gambar berizin. Batas contoh: 20 gambar, maksimum 20 MB per gambar dan 20 megapiksel. Upload notebook memakai memori; jangan unggah berkas raksasa.

```
uploaded = files.upload()
if not uploaded or len(uploaded) > 20:
    raise ValueError('Upload 1–20 gambar.')
images = []
for index, (name, content) in enumerate(uploaded.items()):
    if Path(name).suffix.lower() not in {'.jpg', '.jpeg', '.png'}:
        raise ValueError(f'Format tidak didukung: {name}')
    if not (0 < len(content) <= 20 * 1024**2):
        raise ValueError(f'Ukuran tidak valid: {name}')
    image = cv2.imdecode(np.frombuffer(content, np.uint8),
cv2.IMREAD_COLOR)
    if image is None or image.shape[0] * image.shape[1] >
20_000_000:
        raise ValueError(f'Gambar gagal dibaca atau terlalu besar:
{name}')
    images.append((f'{index:02}_{Path(name).stem}', image))
print('Gambar valid:', len(images))
```

Sel 4 — Pilih objek berdasarkan nama kelas model

Contoh menghitung kendaraan dalam gambar. Ubah TARGET_NAMES menjadi nama yang benar-benar ada di model.names, bukan indeks kelas tebakan.

```
TARGET_NAMES = ['car', 'motorcycle', 'bus', 'truck']
CONF = 0.35
if not (0 < CONF <= 1):
    raise ValueError('Confidence harus di rentang (0,1].')
name_to_id = {name: int(idx) for idx, name in model.names.items()}
unknown = set(TARGET_NAMES) - set(name_to_id)
if unknown or not TARGET_NAMES:
    raise ValueError(f'Nama kelas tidak valid: {unknown}; lihat
model.names.')
TARGET_IDS = [name_to_id[name] for name in TARGET_NAMES]
print('Filter:', dict(zip(TARGET_NAMES, TARGET_IDS)))
```

Sel 5 — Deteksi, koordinat, dan counting per gambar

Filter class diterapkan saat predict; penghitung berasal dari bbox yang benar-benar dikembalikan. Deteksi nol tetap menghasilkan tabel dengan kolom yang konsisten.

```
import pandas as pd
from collections import Counter
from IPython.display import display
all_rows, count_rows = [], []
columns = ['gambar', 'class_id', 'class_name', 'confidence',
           'x1', 'y1', 'x2', 'y2', 'width_px', 'height_px']
for name, image in images:
    result = model.predict(image, conf=CONF, classes=TARGET_IDS,
                           device=DEVICE, imgsz=640, verbose=False)
[0]
    counts = Counter()
    for box in result.boxes:
        class_id = int(box.cls.item())
        class_name = model.names[class_id]
        x1, y1, x2, y2 = box.xyxy[0].cpu().tolist()
        score = float(box.conf.item())
        counts[class_name] += 1
        all_rows.append(dict(gambar=name, class_id=class_id,
            class_name=class_name, confidence=score, x1=x1, y1=y1,
            x2=x2, y2=y2, width_px=x2-x1, height_px=y2-y1))
    count_rows.append({'gambar': name, **{n: counts[n] for n in
TARGET_NAMES},
                       'total_deteksi_target':
sum(counts.values())})
    annotated = result.plot()
    if not cv2.imwrite(str(OUTPUT / f'{name}_count.jpg'),
annotated):
        raise IOError('Gagal menyimpan citra hasil.')
    plt.figure(figsize=(10, 7))
    plt.imshow(cv2.cvtColor(annotated, cv2.COLOR_BGR2RGB))
    plt.title(f'{name} — {dict(counts)}'); plt.axis('off');
plt.show()
detections = pd.DataFrame(all_rows, columns=columns)
count_table = pd.DataFrame(count_rows)
display(detections); display(count_table)
detections.to_csv(OUTPUT / 'bbox.csv', index=False)
count_table.to_csv(OUTPUT / 'counting.csv', index=False)
```

Sel 6 — Masukkan hitungan manual dan hitung galat

Isi saat prompt berdasarkan gambar asli dan kebijakan target yang sama. Nilai manual tidak disiapkan sebagai ground truth buatan.

```
manual_rows = []
for name, image in images:
    plt.figure(figsize=(9, 6))
    plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
    plt.title(f'Hitung manual: {name}'); plt.axis('off'); plt.show()
    raw = input(f'Jumlah manual objek target untuk {name}:
').strip()
    if not raw.isdigit():
        raise ValueError('Masukkan bilangan bulat nonnegatif.')
    manual_rows.append({'gambar': name, 'jumlah_manual': int(raw)})
audit = count_table.merge(pd.DataFrame(manual_rows), on='gambar',
validate='one_to_one')
audit['galat'] = audit['total_deteksi_target'] -
audit['jumlah_manual']
audit['galat_absolut'] = audit['galat'].abs()
display(audit)
print('Rata-rata galat absolut hitungan:',
audit['galat_absolut'].mean())
audit.to_csv(OUTPUT / 'audit_counting.csv', index=False)
```

Jika terjadi error

Filter nama tidak dikenali.

Periksa model.names; gunakan nama persis dari model atau latih custom, bukan membuat ID baru.

Tabel counting menyatakan nol.

Nol adalah hasil sah. Cek objek terlihat, kelas target dan threshold tanpa menambah bbox manual sebagai hasil model.

Galat hitungan nol dianggap sempurna.

Periksa bbox satu per satu karena false positive dan false negative dapat saling meniadakan.

> Runtime Colab dan file upload bersifat sementara. Simpan output penting pada notebook/Drive sebelum sesi berakhir. Jika runtime direset, jalankan ulang sel dari awal.

Gunakan media berizin dan hindari unggahan data sensitif tanpa izin.
