PENGOLAHAN CITRA · MATERI 10

Pengenalan YOLO dan Object Detection

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

Perhatikan foto sebuah jalan. Di dalamnya mungkin ada kendaraan, orang, dan benda lain yang letaknya berbeda-beda. Menyebut satu nama untuk seluruh foto belum cukup jika kita ingin mengetahui objek mana yang berada di kiri atau kanan. Kita membutuhkan keluaran yang menghubungkan nama objek dengan lokasinya. Tugas inilah yang disebut deteksi objek: program mengusulkan kategori dan kotak lokasi untuk objek yang dikenali.

Bab sebelumnya menggunakan klasifikasi, sehingga keluarannya berupa peringkat kategori untuk gambar. Pada bab ini kita menggunakan YOLO untuk melihat beberapa kandidat objek dalam satu foto. Kotak yang muncul disebut bounding box. Bersama kotak itu ada nama kelas dan confidence, yaitu skor model yang membantu menyaring kandidat. Kotak bukan garis bentuk objek yang persis, dan skor tinggi belum membuktikan bahwa prediksinya benar.

Kita memakai bobot pretrained `yolov8n.pt` sebagai titik awal yang ringan. Kata pretrained berarti model sudah dilatih sebelumnya; mengunggah gambar ke Colab tidak langsung melatih ulang model tersebut. Daftar objek yang dapat diprediksi juga mengikuti kelas model, bukan seluruh benda yang mungkin ditemukan di lingkungan kita. Jika suatu benda tidak mendapat kotak, kita perlu memeriksa kategori yang tersedia, kondisi foto, dan pengaturan penyaringan sebelum menyimpulkan penyebabnya.

Urutan pembahasan bergerak dari arti deteksi, cara membaca koordinat, lalu perbedaan confidence dan tumpang tindih kotak. Ilustrasi membantu menghubungkan kotak dengan bagian foto, sedangkan video memberi gambaran cara kerja deteksi. Praktik Colab kemudian menjalankan inferensi yang sebenarnya, menampilkan anotasi, dan menyimpan ringkasannya. Fokusnya bukan memperoleh gambar yang terlihat ramai oleh kotak, melainkan dapat menjelaskan dari mana setiap informasi berasal dan apa batas kesimpulannya.

2. Pahami konsep secara berurutan

2.1 Deteksi menghubungkan kategori dengan lokasi

Object detection atau deteksi objek menghasilkan kandidat objek beserta kategori dan lokasinya. Satu foto dapat mempunyai beberapa kandidat dari kelas yang sama maupun kelas berbeda. Bounding box adalah persegi panjang yang mewakili perkiraan lokasi tersebut. Kotak sengaja merupakan bentuk sederhana: ia dapat memuat latar di sekitar objek dan tidak menggambarkan lekuk objek secara tepat. Karena itu, deteksi berbeda dari segmentasi yang bekerja pada wilayah piksel.

YOLO merupakan keluarga model deteksi. Praktikum ini memakai `yolov8n.pt`, model pretrained berukuran nano sebagai titik awal yang ringan. Model telah mempelajari pola dari data berlabel sebelum kita memakainya. Saat kita menjalankan inferensi, bobot tidak diperbarui dengan foto unggahan. Pemilihan model tersebut juga bukan pernyataan bahwa ia paling baru atau paling tepat untuk semua kebutuhan. Keluaran perlu diperiksa pada kondisi foto yang benar-benar digunakan.

2.2 Baca kotak dengan sistem koordinat yang benar

Dalam gambar digital, posisi horizontal x bertambah ke kanan dan posisi vertikal y bertambah ke bawah. Bentuk `xyxy` menyimpan titik kiri atas, yaitu x1 dan y1, kemudian titik kanan bawah, yaitu x2 dan y2. Lebar kotak diperoleh dari selisih x2 dengan x1, sedangkan tingginya dari selisih y2 dengan y1. Koordinat `Results.boxes.xyxy` mengacu pada gambar asli sehingga dapat dibaca bersama dimensi foto tersebut.

Jangan menukar bentuk ini dengan `xywh`, yang menyatakan pusat kotak, lebar, dan tinggi. Koordinat piksel juga berbeda dari koordinat ternormalisasi yang menyatakan posisi relatif terhadap ukuran gambar. Kesalahan membaca format dapat memindahkan kotak atau menghasilkan ukuran yang keliru walaupun model berjalan normal. Sebuah kotak yang tampak besar pun tidak otomatis berarti objek fisiknya besar, sebab jarak kamera dan perspektif memengaruhi ukuran pada foto.

2.3 Pisahkan nama kelas dari skor confidence

Setiap deteksi mempunyai ID kelas. Pemetaan `model.names` menghubungkan ID tersebut dengan nama yang dapat dibaca. Gunakan pemetaan milik model yang sedang dimuat, bukan angka yang dihafalkan dari contoh lain. Bobot pretrained mempunyai kosakata terbatas. Ia tidak otomatis mengenali kategori produk lokal, kerusakan kendaraan, atau semua spesies tanaman. Benda yang tidak termasuk kosakata dapat diabaikan atau justru diberi nama kelas yang keliru.

Confidence adalah skor kandidat, bukan ukuran ketepatan keseluruhan model. Parameter `conf` menetapkan batas skor untuk penyaringan. Menaikkan batas dapat membuang kandidat

lemah, termasuk objek benar yang kecil atau tertutup. Menurunkannya dapat menambah kandidat berguna sekaligus prediksi keliru. Ini pertukaran yang perlu diamati, bukan resep bahwa satu nilai selalu terbaik. Ketika membandingkan hasil, pertahankan gambar dan pengaturan lain agar perubahan lebih mudah dikaitkan dengan confidence.

2.4 Pahami IoU sebagai ukuran tumpang tindih

Intersection over Union atau IoU mengukur perbandingan luas irisan dua kotak terhadap luas gabungannya. Luas gabungan diperoleh dengan menjumlahkan luas kedua kotak lalu mengurangi bagian irisan yang terhitung dua kali. Kotak yang tidak saling beririsan memiliki IoU nol, sedangkan dua kotak identik dengan luas positif memiliki IoU satu. Nilai ini membicarakan hubungan lokasi kotak, bukan confidence atau nama kelas.

Untuk evaluasi, sebuah prediksi dapat dibandingkan dengan kotak rujukan yang dibuat pengamat. Akan tetapi, dua kotak buatan di notebook hanya menjelaskan geometri, bukan mengukur ketepatan model terhadap anotasi manusia. Fungsi `box_iou` memeriksa bahwa setiap kotak mempunyai luas positif sebelum menghitung. Memisahkan latihan geometri dari keluaran inferensi mencegah angka ilustratif disalahpahami sebagai hasil pengujian detektor.

2.5 Bedakan penyaringan kotak dari evaluasi ketepatan

Satu objek dapat memunculkan beberapa kandidat kotak yang mirip. Pada alur baseline ini, non-maximum suppression atau NMS membantu menekan kandidat yang saling bertumpuk menurut aturan skornya. Parameter `iou` pada pemanggilan `predict` mengatur ambang tumpang tindih untuk proses tersebut. Ia bukan target akurasi dan bukan nilai IoU yang diukur terhadap anotasi manusia.

Ambang NMS lebih rendah dapat menekan kotak yang tumpang tindih lebih kuat. Ini dapat mengurangi kandidat berlebih, tetapi juga perlu dicermati ketika objek berbeda berdiri berdekatan. Confidence dan IoU dengan demikian menjawab hal berbeda: skor untuk mempertahankan kandidat, serta hubungan lokasi antarkotak. Praktikum mencetak jumlah deteksi, tetapi jumlah itu sendiri belum menunjukkan berapa kandidat yang benar. Penilaian ketepatan memerlukan rujukan dan pemeriksaan yang tidak disediakan otomatis oleh banyaknya kotak.

2.6 Ikuti alur gambar sampai keluaran yang dapat diperiksa

Kode membaca unggahan memakai OpenCV sebagai array BGR. Setelah `model.predict` dijalankan, `result.plot()` membuat gambar anotasi dalam urutan warna BGR. Matplotlib memerlukan konversi ke RGB agar warna tampil semestinya. Sebaliknya, `cv2.imwrite` menyimpan array BGR secara langsung. Kesalahan urutan warna dapat membuat tampilan aneh; jangan langsung menganggapnya kesalahan pengenalan objek.

Setiap gambar menghasilkan anotasi dan baris ringkasan dengan jumlah deteksi serta threshold. Jika tidak ada deteksi, nol adalah keluaran yang sah. Periksa foto asli, kelas model, ukuran objek, cahaya, dan pengaturan tanpa menambahkan kotak buatan seolah hasil inferensi. Berkas di direktori runtime bersifat sementara, sehingga output penting perlu disimpan. Gunakan media berizin, hindari foto sensitif tanpa persetujuan, dan jangan mengubah hasil latihan ini menjadi klaim keselamatan atau ketepatan umum yang belum diuji.



Inferensi aktual YOLOv8n pada foto sumber; tidak menjamin semua objek terdeteksi.

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

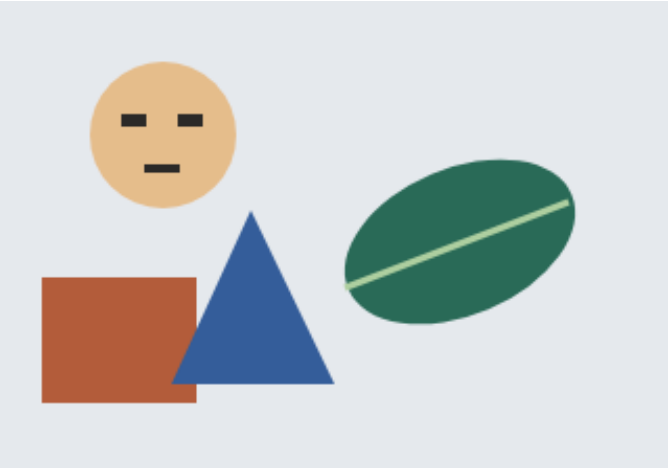
Ilustrasi kotak manual: IoU dan confidence

Dua kotak geometri manual di atas foto, bukan hasil YOLO. A tetap; B digeser dengan koordinat. Ambang hanya menyaring skor buatan B, tidak mengubah IoU.

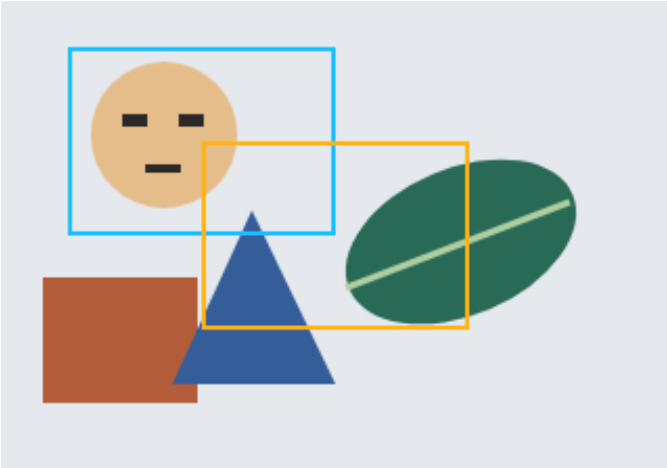
> Kotak dan skor yang ditampilkan di sini adalah contoh manual untuk mempelajari geometri dan penyaringan, bukan hasil prediksi YOLO terhadap foto Anda. Perhitungan IoU berasal dari koordinat contoh yang ditampilkan.

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: Koordinat kiri B (% lebar foto) = 30 · Koordinat atas B (% tinggi foto) = 30 · Skor B manual (bukan hasil model) = 0.7 · Ambang skor ilustratif B = 0.5

Ilustrasi manual: IoU = 0.1429; 1 kandidat B lolos skor ≥ 0.50. Bukan deteksi foto.

A biru = acuan geometri manual; B jingga = lolos, B abu-abu putus-putus = tersaring (tetap ditampilkan untuk latihan IoU). Semua kotak berukuran 40%×40% gambar. Irisan = 2867.20 px²; gabungan = 20070.40 px²; IoU = irisan/gabungan. Skor B 0.70 ditetapkan manual, bukan confidence YOLO atau kepastian benar. A tidak ikut filter. Tidak ada inferensi, NMS, ground truth, atau pengukuran akurasi.

▼ **Angka dan parameter hasil perhitungan ilustrasi**


```
{
  "illustration": true,
  "inference": false,
  "boxA": [
    32,
    22.400000000000002,
    160,
    112
  ],
  "boxB": [
    96,
    67.2,
```

```
    224,
    156.79999999999998
  ],
  "areaA": 11468.800000000003,
  "areaB": 11468.799999999997,
  "intersectionArea": 2867.2000000000007,
  "unionArea": 20070.4,
  "iou": 0.14285714285714288,
  "manualScore": 0.7,
  "threshold": 0.5,
  "retainedCandidateCount": 1
}
```


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

How YOLO Object Detection Works — DeepBean.

Fokus saat menonton: Amati ide deteksi sekali lintas, class dan bounding box. Video menjelaskan konsep YOLO; arsitektur YOLO generasi awal bukan rincian identik YOLOv8.

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
- Google Colab — FAQ dan keterbatasan runtime
- Matplotlib — imshow

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 Ultralytics, kemudian sel pemuatan `YOLO('yolov8n.pt')`. Periksa versi, `DEVICE`, dan `model.names` yang tercetak sebelum mengunggah foto. Sel validasi menerima 1–20 JPG/PNG dengan batas 20 MB per berkas dan 20 megapiksel per gambar. Gambar dibaca menjadi array BGR lalu disimpan dalam `images`; berkas yang gagal dibaca atau melampaui batas menghentikan sel dengan pesan kesalahan.

Sel inferensi memproses setiap gambar dengan `conf=0.25`, `iou=0.7`, dan `imgsz=640`. Keluaran `result` dipakai untuk menghitung kotak dan menggambar anotasi. Periksa nama kelas serta lokasi kotaknya, bukan jumlahnya saja. Gambar anotasi disimpan melalui `OUTPUT` ke `/content/hasil_yolo`, sedangkan `ringkasan.csv` memuat nama gambar, jumlah deteksi, threshold, dan lokasi hasil. Bila mencoba confidence lain, bandingkan gambar yang sama dan ingat bahwa menjalankan ulang sel dapat menimpa berkas keluaran sebelumnya.

Sel `box_iou` berdiri sebagai latihan geometri dengan kotak `a` dan `b` yang sudah dinyatakan buatan. Hasilnya bukan evaluasi model terhadap foto unggahan. Hubungkan perhitungan ini dengan pembahasan tumpang tindih, lalu kembali amati keluaran deteksi aktual. Jika warna tampak salah, periksa konversi BGR ke RGB untuk tampilan. Jika tidak ada kotak, baca hasil nol apa adanya dan periksa kategori serta kondisi foto sebelum menarik kesimpulan.

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 — Inferensi, tampilkan, dan simpan semua gambar

Jalankan setelah upload. Tiap gambar mendapat output anotasi dan ringkasan jumlah deteksi. Periksa hasil satu gambar terlebih dahulu sebelum menambah masukan.

```
summary = []
for name, image in images:
    result = model.predict(image, conf=0.25, iou=0.7, imgsz=640,
                           device=DEVICE, verbose=False)[0]
    annotated = result.plot()
    target = OUTPUT / f'{name}_deteksi.jpg'
    if not cv2.imwrite(str(target), annotated):
        raise IOError(f'Gagal menyimpan {target}')
    summary.append({'gambar': name, 'jumlah_deteksi':
len(result.boxes),
                    'conf_threshold': 0.25, 'hasil': str(target)})
    plt.figure(figsize=(10, 7))
    plt.imshow(cv2.cvtColor(annotated, cv2.COLOR_BGR2RGB))
    plt.title(f'{name}: {len(result.boxes)} deteksi')
    plt.axis('off'); plt.show()
import pandas as pd
from IPython.display import display
display(pd.DataFrame(summary))
pd.DataFrame(summary).to_csv(OUTPUT / 'ringkasan.csv', index=False)
```

Sel 5 — Menghitung IoU dua kotak ilustratif

Angka di sini adalah koordinat buatan untuk latihan geometri, bukan keluaran inferensi.

```
def box_iou(a, b):
    for box in (a, b):
        if len(box) != 4 or box[2] <= box[0] or box[3] <= box[1]:
            raise ValueError('Kotak harus xyxy dengan luas
positif.')
    iw = max(0, min(a[2], b[2]) - max(a[0], b[0]))
    ih = max(0, min(a[3], b[3]) - max(a[1], b[1]))
    intersection = iw * ih
    area_a = (a[2] - a[0]) * (a[3] - a[1])
    area_b = (b[2] - b[0]) * (b[3] - b[1])
    return intersection / (area_a + area_b - intersection)
a = [10, 10, 110, 110]
b = [60, 60, 160, 160]
print('IoU kotak ilustratif:', box_iou(a, b))
assert box_iou(a, a) == 1
assert box_iou(a, [200, 200, 210, 210]) == 0
```

Jika terjadi error

Bobot gagal diunduh.

Periksa koneksi runtime dan ulangi pemuatan yolov8n.pt; jangan mengganti hasil dengan data buatan.

Warna hasil salah.

Konversikan hasil plot BGR ke RGB sebelum Matplotlib, bukan sebelum cv2.imwrite.

Gambar tidak memiliki deteksi.

Periksa kelas model dan threshold; laporkan hasil nol apa adanya.

> 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.
