PENGOLAHAN CITRA · MATERI 09

# Pengenalan Artificial Intelligence untuk Computer Vision

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 membuka foto sebuah kendaraan di telepon. Kita dapat menyebutnya bus karena mengenali bentuk, roda, jendela, dan konteks jalan. Pada materi sebelumnya, program mengolah piksel melalui aturan yang kita tentukan. Misalnya, warna tertentu dipilih dengan ambang batas. Pendekatan itu berguna, tetapi aturan warna saja belum cukup untuk membedakan bus dari benda lain yang kebetulan berwarna sama. Di sinilah pembelajaran dari contoh menjadi menarik: sebagian pola pengenal tidak ditulis satu per satu, melainkan dipelajari saat model dilatih.

Bab ini memperkenalkan artificial intelligence atau AI untuk membaca gambar melalui tugas yang sederhana, yaitu klasifikasi. Satu foto menjadi masukan, kemudian model memberikan peringkat nama kategori beserta skornya. Kita menggunakan MobileNetV3 Small yang sudah dilatih sebelumnya. Foto yang diunggah dipakai untuk mencoba model, bukan untuk mengajarinya kategori baru. Karena itu, menjalankan praktikum tidak sama dengan membuat model yang khusus mengenali benda di lingkungan kita.

Bayangkan pula foto yang memuat bus, pohon, dan pejalan kaki. Classifier dalam bab ini tidak menunjukkan letak masing-masing objek. Ia memberi skor kategori untuk gambar sebagai satu masukan. Nama yang muncul belum tentu menggambarkan objek yang ingin kita periksa. Apabila kategori yang kita butuhkan tidak tersedia, model tetap dapat memilih kategori lain dengan skor tinggi. Skor tersebut perlu dibaca sebagai keluaran model, bukan jaminan kebenaran.

Alur belajar dimulai dari hubungan AI, machine learning, dan deep learning, lalu beralih ke cara gambar disiapkan dan prediksi dibaca. Setelah mengamati ilustrasi dan video, kita menjalankan alur yang sama di Colab: memeriksa foto, menetapkan label pengamatan, memperoleh tiga prediksi teratas, dan membandingkannya secara jujur. Yang dipelajari bukan sekadar menjalankan program, melainkan memahami apa yang sebenarnya dapat disimpulkan dari keluarannya.

# 2. Pahami konsep secara berurutan

## 2.1 Dari aturan buatan menuju pola yang dipelajari

Artificial intelligence adalah istilah luas untuk sistem yang melakukan tugas yang dikaitkan dengan kemampuan cerdas. Machine learning merupakan salah satu pendekatannya: program mempelajari pola dari contoh untuk membantu menghasilkan keputusan. Dalam pengolahan citra, aturan manual dapat berbunyi memilih piksel hijau. Model pembelajaran bekerja berbeda karena parameter pengenalannya dibentuk melalui data dan proses pelatihan. Perbedaan ini bukan berarti aturan manual selalu buruk atau model pembelajaran selalu lebih tepat. Keduanya harus dipilih sesuai persoalan.

Deep learning adalah bagian machine learning yang memakai jaringan dengan banyak tahap pengolahan. CNN, singkatan dari convolutional neural network, merupakan keluarga jaringan yang mengolah pola lokal pada citra. Secara intuitif, informasi sederhana seperti tepi dan tekstur dapat digabungkan menjadi representasi yang membantu pengenalan. Ini gambaran untuk memahami alurnya, bukan penjelasan pasti tentang arti setiap neuron. Jaringan tetap dapat memanfaatkan latar yang kebetulan berkaitan dengan label, sehingga penampilannya pada foto baru perlu diperiksa.

## 2.2 Tentukan dahulu tugas yang dikerjakan model

Klasifikasi gambar memberikan skor kategori untuk satu gambar. Deteksi objek menambahkan lokasi, biasanya berupa kotak untuk beberapa objek. Segmentasi lebih rinci lagi karena membedakan wilayah pada tingkat piksel. Perbedaan keluaran ini penting: menempelkan tulisan prediksi di atas foto tidak membuat classifier berubah menjadi detector. MobileNet pada praktikum ini tidak mengembalikan koordinat bus, pohon, atau orang.

Untuk memahami klasifikasi, mulai dari foto dengan satu objek utama yang terlihat jelas. Sesudah itu, foto dengan banyak objek membantu menunjukkan batasannya. Pada foto meja berisi beberapa benda, misalnya, kategori teratas dapat berkaitan dengan benda yang bukan fokus pengamat. Contoh ini bersifat konseptual, bukan hasil prediksi yang sudah dijalankan. Tetapkan objek yang ingin dinilai sebelum membaca keluaran agar penilaian tidak bergeser mengikuti jawaban model.

## 2.3 Bedakan pelatihan, bobot pretrained, dan inferensi

Pelatihan adalah proses menyesuaikan parameter model menggunakan contoh. Parameter yang sudah diperoleh dapat disimpan sebagai bobot. Istilah pretrained berarti bobot tersebut berasal dari pelatihan sebelumnya. Inferensi adalah penggunaan model dan bobot itu untuk memproses masukan baru. Pada praktikum, mengunggah foto lalu menjalankan model hanya melakukan inferensi; tidak ada langkah pembaruan bobot berdasarkan foto mahasiswa.

Kode menggunakan `mobilenet_v3_small` dengan `MobileNet_V3_Small_Weights.IMAGENET1K_V1`. Nama kategori diambil dari `weights.meta['categories']`, sehingga urutannya mengikuti keluaran bobot yang sama. Pemilihan model kecil ini dimaksudkan sebagai titik awal yang ringan, bukan klaim model terbaru atau paling unggul. Bobot perlu diunduh saat pertama digunakan. Apabila unduhan gagal, proses belum menghasilkan prediksi yang sah; jangan menggantinya dengan nama kategori rekaan.

## 2.4 Siapkan gambar sesuai cara model dilatih

Foto yang terlihat oleh manusia belum langsung menjadi masukan jaringan. Kode membetulkan orientasi berdasarkan informasi gambar, mengubahnya menjadi RGB, lalu memakai `weights.transforms()`. Transformasi resmi mencakup pengubahan ukuran, pemotongan, dan normalisasi yang sesuai dengan bobot. Normalisasi mengatur nilai masukan menurut kebutuhan model, bukan menjamin bahwa pencahayaan atau kualitas foto sudah ideal.

Pemotongan pusat dapat menghilangkan objek di tepi. Karena itu, foto utuh yang ditampilkan notebook belum tentu identik dengan wilayah yang masuk ke jaringan setelah transformasi. Ini salah satu penjelasan yang perlu diperiksa ketika label tampak tidak sesuai, bukan alasan untuk membenarkan semua kesalahan. `model.eval()` mengatur perilaku modul tertentu ke mode evaluasi. Sementara itu, `torch.no_grad()` mematikan pencatatan gradien yang tidak dibutuhkan dalam inferensi. Keduanya berbeda fungsi dan tidak meningkatkan kebenaran prediksi secara otomatis.

## 2.5 Baca top-3 sebagai peringkat dalam kosakata terbatas

Keluaran awal jaringan disebut logits. Fungsi softmax mengubahnya menjadi skor relatif untuk seluruh kategori yang tersedia. Kode menampilkan tiga skor terbesar melalui `torch.topk`. Top-1 berarti kategori di urutan pertama, sedangkan top-3 mencakup tiga kategori teratas. Melihat alternatif membantu mengamati persaingan antarkategori, tetapi tidak mengubah skor menjadi ukuran akurasi.

Classifier ini memiliki kosakata tertutup: ia memilih dari kategori yang sudah tersedia, bukan menciptakan kategori baru. Foto daun lokal tidak otomatis dapat diberi nama spesies atau diagnosis penyakit. Skor tinggi juga bukan bukti bahwa objek tersebut benar-benar termasuk salah satu kelas model. Karena itu, label pengamatan `DI_LUAR_KELAS` dipakai ketika kategori yang diperlukan tidak ada, sedangkan `TIDAK_PASTI` dipakai ketika pengamat belum dapat menetapkan label. Keduanya adalah catatan manusia, bukan kategori penolakan yang diprediksi model.

## 2.6 Bandingkan hasil tanpa melebihkan kesimpulan

Label pengamatan perlu ditentukan sebelum inferensi. Untuk kategori yang tersedia, ringkasan menghitung apakah label itu cocok dengan top-1 atau muncul dalam top-3. Gambar bertanda khusus dipisahkan dari perhitungan tersebut. Pemisahan ini menjaga agar kategori yang tidak tersedia tidak diperlakukan seperti kategori biasa. Namun, kualitas ringkasan tetap bergantung pada ketepatan label manusia dan pilihan foto.

Kumpulan kecil dalam praktikum adalah bahan pengamatan, bukan ukuran akurasi umum model. Foto duplikat atau potongan dari objek yang sama juga tidak memberi keragaman setara dengan objek berbeda. Kondisi cahaya, sudut pandang, latar, dan jenis objek dapat berbeda dari data pelatihan dan menimbulkan kegagalan. Catat ketidakcocokan yang benar-benar terlihat, termasuk ketidakpastian pengamat. Gunakan gambar berizin dan hindari data sensitif; keluaran latihan tidak layak menjadi dasar diagnosis, penetapan identitas, atau keputusan keselamatan.




Inferensi aktual ImageNet; skor relatif kelas model, bukan kepastian jenis objek.

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

---

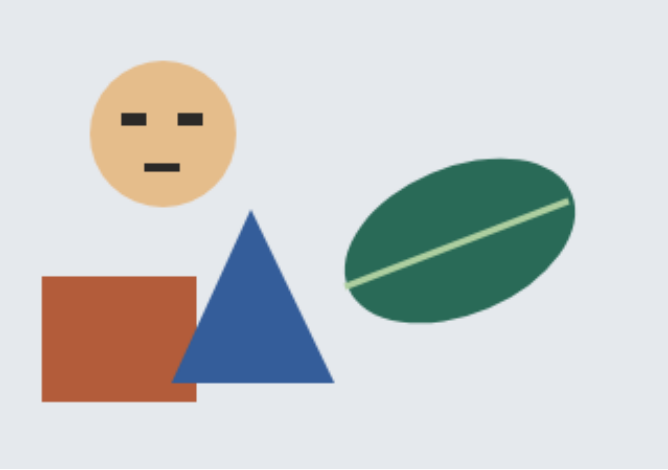
# 3. Ilustrasi interaktif sebelum praktikum

## Ilustrasi preprocessing dan softmax manual

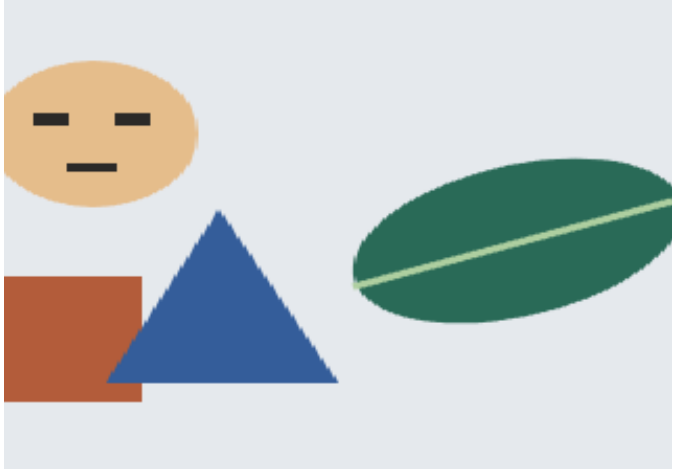
Piksel unggahan benar-benar diproses menjadi 224×224. Softmax hanya memakai logit manual A, B, C; bukan klasifikasi foto atau inferensi MobileNet.

> Ilustrasi ini membantu memahami prapemrosesan dan angka yang dipakai dalam alur AI, bukan memberikan prediksi kelas foto dari model pretrained. Inferensi model dengan bobot resmi tetap dilakukan di Colab.

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: Geometri 224×224 (ilustrasi) = crop · Kanal pratinjau = rgb · Logit A manual (tidak berasal dari foto) = 1 · Logit B manual; logit C = 0 = 0

Ilustrasi: crop tengah → 224×224 RGB; bukan prediksi AI.

Sampling nearest-neighbor sederhana, bukan transforms resmi MobileNetV3. Pratinjau 224×224 dipetakan kembali ke 320×224 untuk tampilan (rasio dapat berubah). Statistik dihitung pada piksel 224×224: mean kanal [0,1] = 0.7509, 0.7746, 0.7700; mean (x−mean)/std = 1.1613, 1.4222, 1.6176. Normalisasi bersifat numerik; tampilan tetap piksel RGB/grayscale, bukan nilai tensor. Softmax manual A/B/C = 0.5761, 0.2119, 0.2119 dari logit 1, 0, 0; label A/B/C hanyalah simbol latihan, bukan kelas foto. Praktikum inferensi tetap memerlukan bobot, transforms resmi, eval() dan no_grad() di Colab. Skor bukan kepastian benar.

▼ **Angka dan parameter hasil perhitungan ilustrasi**


```
{
  "illustration": true,
  "inference": false,
  "preprocessingSize": [
    224,
    224
  ],
  "sampling": "nearest-neighbor",
  "geometry": "crop",
  "channelMeans": [
    0.7509389693373965,
    0.7745779561818996,
    0.7699618910064898
  ],
  "normalizedMeans": [
    1.1613055429580632,
    1.4222230186691946,
    1.617608404473288
  ],
  "logits": [
    1,
    0,
    0
  ],
  "probabilities": [
    0.5761168847658291,
    0.21194155761708544,
    0.21194155761708544
  ],
  "scoreSource": "manual"
}
```


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

**But what is a neural network? | Deep learning chapter 1** — 3Blue1Brown.

**Fokus saat menonton:** Amati neuron, bobot, aktivasi dan pembelajaran. Hubungkan dengan CNN/pretrained di bab ini; video menjelaskan dasar jaringan, bukan API MobileNet.

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**

- torchvision — MobileNetV3 Small dan preprocessing bobot
- torchvision — Models and pre-trained weights
- PyTorch — Module.eval
- PyTorch — no_grad

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.

Mulai dengan sel import, lalu muat bobot `IMAGENET1K_V1` dan periksa `classes` serta `preprocess` yang tercetak. Gunakan setup opsional hanya jika paket belum tersedia. Sel unggah menerima 10–20 foto JPG/PNG, masing-masing maksimal 5 MB, total maksimal 50 MB, dan sisi maksimal 4096 piksel. Pemeriksaan dapat menolak berkas tertentu; alur baru dapat dilanjutkan ketika sedikitnya sepuluh gambar valid tersimpan pada `images`. Gunakan foto berbeda yang jelas sumber izinnya.

Sebelum inferensi, amati setiap foto dan cari nama kategori pada `classes`. Prompt menyimpan label itu sebagai `expected`; gunakan `DI_LUAR_KELAS` atau `TIDAK_PASTI` sesuai kondisi, bukan memaksakan nama yang tampak mirip. Kode memerlukan sedikitnya lima

gambar dengan kategori tersedia. Setelah label ditetapkan, jalankan transformasi dan inferensi. Variabel `results` menyimpan top-3 aktual, kemudian notebook menampilkan foto beserta nama kategori, skor, dan label pengamatan sebelumnya.

Baca ringkasan `hits_top1` dan `hits_top3` hanya untuk `known_results`, lalu cocokkan kembali dengan foto yang ditampilkan. Perhatikan perbedaan antara label salah, objek di luar kosakata, dan gambar yang memang ambigu. Bila skor tinggi tampak tidak masuk akal, periksa objek utama, konteks, dan kemungkinan crop sebelum menarik kesimpulan. Simpan output yang ingin dipelajari kembali karena berkas unggahan dan keadaan runtime dapat hilang ketika sesi Colab berakhir.

## Sel 1 — 1. Import

Runtime Colab standar umumnya menyediakan PyTorch/torchvision. Bila import tidak tersedia, jalankan sel setup opsional lalu restart runtime bila diminta.

```
import io, math, warnings
import torch, torchvision
import matplotlib.pyplot as plt
from PIL import Image, ImageOps
from google.colab import files
from torchvision.models import mobilenet_v3_small,
MobileNet_V3_Small_Weights
print('torch:', torch.__version__, '| torchvision:',
torchvision.__version__)
```

## Sel 2 — 2. Muat bobot dan label resmi

Unduhan pertama perlu jaringan. CPU menjadi pilihan eksplisit untuk demo kecil; bobot tidak dilatih ulang.

```
device = torch.device('cpu')
weights = MobileNet_V3_Small_Weights.IMAGENET1K_V1
preprocess = weights.transforms()
model = mobilenet_v3_small(weights=weights).to(device)
model.eval()
classes = weights.meta['categories']
class_lookup = {label.casefold(): label for label in classes}
print('Jumlah kategori model:', len(classes))
print('Contoh pencarian label mengandung cat:', [x for x in classes
if 'cat' in x.lower()][:20])
print('Untuk mencari kelas lain: [x for x in classes if "bus" in
```

```
x.lower()]')
print(preprocess)
```

## Sel 3 — 3. Unggah minimal sepuluh gambar

Pilih 10–20 JPG/PNG berbeda, masing-masing maksimal 5 MB, total maksimal 50 MB, sisi maksimal 4096 piksel. Masukan yang ditolak dicetak; minimal 10 gambar valid wajib tersedia.

```
uploaded = files.upload()
if not 10 <= len(uploaded) <= 20:
    raise ValueError('Unggah 10 sampai 20 gambar, bukan ZIP.')
if sum(len(raw) for raw in uploaded.values()) > 50 * 1024 * 1024:
    raise ValueError('Total unggahan maksimal 50 MB.')
images = []
for nama, raw in uploaded.items():
    try:
        if not nama.lower().endswith(('.jpg', '.jpeg', '.png')) or
len(raw) > 5 * 1024 * 1024:
            raise ValueError('Format/ukuran berkas melampaui
batas.')
        with warnings.catch_warnings():
            warnings.simplefilter('error',
Image.DecompressionBombWarning)
            with Image.open(io.BytesIO(raw)) as source:
                if max(source.size) > 4096:
                    raise ValueError('Sisi gambar maksimal 4096
piksel.')
                if source.format not in ('JPEG', 'PNG'):
                    raise ValueError('Isi berkas bukan JPEG/PNG.')
                pil = ImageOps.exif_transpose(source).convert('RGB')
                pil.load()
        images.append({'name': nama, 'image': pil})
    except Exception as exc:
        print('Ditolak:', nama, '|', str(exc))
if len(images) < 10:
    raise ValueError('Gambar valid kurang dari 10; unggah ulang
kumpulan yang sesuai.')
print('Gambar valid:', len(images))
for item in images:
    plt.figure(figsize=(4, 3))
```

```
    plt.imshow(item['image']); plt.title(item['name']);
plt.axis('off'); plt.show()
```

## Sel 4 — 4. Tetapkan label pengamatan sebelum inferensi

Ketik nama kategori persis dari classes. Gunakan DI_LUAR_KELAS untuk objek yang kelasnya tidak tersedia, atau TIDAK_PASTI jika ambigu. Minimal lima gambar harus dapat diberi label kelas tersedia.

```
special = {'DI_LUAR_KELAS', 'TIDAK_PASTI'}
for item in images:
    while True:
        answer = input(f"Label pengamatan untuk {item['name']}:
").strip()
        if answer.upper() in special:
            item['expected'] = answer.upper()
            break
        if answer.casefold() in class_lookup:
            item['expected'] = class_lookup[answer.casefold()]
            break
        print('Label tidak ada dalam kosakata. Cari di classes atau
gunakan penanda khusus.')
known_count = sum(item['expected'] not in special for item in
images)
if known_count < 5:
    raise ValueError('Sertakan minimal lima gambar dengan kelas
tersedia; jalankan ulang upload dan label.')
print('Label kelas tersedia:', known_count)
```

## Sel 5 — 5. Inferensi top-3 dan visualisasi

Gambar diproses satu per satu agar ringan. Skor softmax adalah keluaran model, bukan kepastian bahwa label benar.

```
results = []
with torch.no_grad():
    for item in images:
        tensor = preprocess(item['image']).unsqueeze(0).to(device)
        logits = model(tensor)[0]
        probs = torch.softmax(logits, dim=0)
        scores, indices = torch.topk(probs, k=3)
```

```
        top3 = [(classes[int(index)], float(score))
                for index, score in zip(indices.cpu(),
scores.cpu())]
        results.append({'name': item['name'], 'expected':
item['expected'], 'top3': top3})
for item, result in zip(images, results):
    plt.figure(figsize=(6, 4))
    plt.imshow(item['image']); plt.axis('off')
    lines = [f"{label}: skor {score:.3f}" for label, score in
result['top3']]
    plt.title(item['name'] + chr(10) + chr(10).join(lines))
    plt.show()
    print('Pengamatan sebelum inferensi:', result['expected'])
```

## Sel 6 — 6. Ringkasan evaluasi kecil

Hanya gambar dengan label tersedia masuk hitungan kesesuaian. Penanda khusus tetap dilaporkan dan tidak disamakan dengan label classifier.

```
known_results = [r for r in results if r['expected'] not in special]
hits_top1 = sum(r['top3'][0][0] == r['expected'] for r in
known_results)
hits_top3 = sum(r['expected'] in [label for label, score in
r['top3']] for r in known_results)
print('Total inferensi:', len(results))
print('Dengan kelas tersedia:', len(known_results))
print('DI_LUAR_KELAS:', sum(r['expected'] == 'DI_LUAR_KELAS' for r
in results))
print('TIDAK_PASTI:', sum(r['expected'] == 'TIDAK_PASTI' for r in
results))
if known_results:
    print(f'Kesesuaian top-1: {hits_top1}/{len(known_results)}
({hits_top1/len(known_results):.1%})')
    print(f'Kesesuaian top-3: {hits_top3}/{len(known_results)}
({hits_top3/len(known_results):.1%})')
print('Ringkasan ini hanya berlaku untuk sampel praktik, bukan
akurasi umum model.')
for r in results:
    print(r['name'], '| pengamatan:', r['expected'], '| top-3:',
r['top3'])
```

## Sel 7 — Setup opsional jika paket belum tersedia

Jalankan hanya jika sel import menghasilkan ModuleNotFoundError. Tidak perlu menjalankan pada Colab yang sudah memiliki pasangan torch/torchvision kompatibel. Setelah instalasi, restart runtime jika paket sudah diimport dan jalankan sel 1 kembali.

```
%pip install torch torchvision
```

## Jika terjadi error

**Unduhan bobot gagal.**

Periksa jaringan runtime dan ulangi pemuatan bobot. Jangan membuat hasil prediksi palsu atau memakai model tanpa bobot tanpa menjelaskan perubahan.

**torchvision tidak cocok dengan torch atau import operator gagal.**

Gunakan runtime Colab baru dengan pasangan paket bawaan kompatibel; jika paket tidak tersedia gunakan setup opsional untuk memasang pasangan bersama, kemudian restart.

**Label pengamatan tidak ada di classes.**

Cari kosakata resmi; bila objek memang tidak memiliki kategori, tandai DI_LUAR_KELAS. Jangan menambahkan nama kategori seolah keluaran model.

**Skor tinggi tetapi gambar tampak tidak cocok.**

Catat sebagai kegagalan/unknown, periksa crop dan konteks gambar, serta jelaskan skor relatif bukan kepastian.

**Jumlah gambar valid kurang dari sepuluh.**

Periksa pesan penolakan, ganti berkas bermasalah, dan unggah ulang 10–20 gambar yang memenuhi batas.

> 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.

---