

Judul Modul: Sistem Deteksi Kecacatan Permukaan Rel Kereta Api Berbasis YOLOv8n dan IP Camera

Tujuan Pembelajaran: Setelah menyelesaikan tugas ini, mahasiswa diharapkan mampu:

1. Memahami konsep dasar deteksi objek dan arsitektur YOLOv8.
2. Mengumpulkan dan menganotasi dataset gambar permukaan rel kereta api dengan berbagai jenis kecacatan.
3. Melakukan transfer learning menggunakan model YOLOv8n yang telah dilatih sebelumnya.
4. Melatih model YOLOv8n dengan dataset yang telah diannotasi.
5. Mengintegrasikan model YOLOv8n dengan input dari IP camera secara *real-time* atau *near real-time*.
6. Mengevaluasi kinerja model deteksi objek menggunakan metrik yang relevan.
7. Mendemonstrasikan sistem deteksi kecacatan yang telah dibangun.

Materi Pendukung:

- Materi kuliah tentang deteksi objek dan arsitektur YOLO.
- Dokumentasi resmi YOLOv8 (<https://docs.ultralytics.com/>).
- Tutorial penggunaan library Python untuk pengolahan citra (misalnya OpenCV, Pillow).
- Contoh dataset permukaan rel kereta api (jika tersedia, atau mahasiswa dapat diminta untuk mencarinya).
- Referensi jurnal atau artikel ilmiah terkait deteksi cacat pada infrastruktur menggunakan visi komputer.

Tugas:

Mahasiswa dibagi menjadi kelompok (maksimal 3-4 orang per kelompok). Setiap kelompok bertugas untuk:

Tahap 1: Pengumpulan dan Anotasi Data

1. Pengumpulan Data:

- Mencari dan mengumpulkan gambar permukaan rel kereta api yang menunjukkan berbagai jenis kecacatan (peeling, corrosion, scratch, crack, corrugation). Sumber data dapat berupa dataset publik (jika ada), gambar dari internet, atau bahkan simulasi gambar.
- Mengumpulkan juga gambar permukaan rel yang tidak mengalami kecacatan (sebagai kelas "normal").
- Jumlah gambar yang dikumpulkan minimal [tentukan jumlah minimum yang realistis, misalnya 200-300 gambar per kelas].

2. Anotasi Data:

- Menggunakan tools anotasi gambar (misalnya LabelImg, Roboflow) untuk memberikan *bounding box* pada setiap objek kecacatan dalam gambar dan melabelinya sesuai dengan jenis kecacatannya (peeling, corrosion, scratch, crack, corrugation).
 - Memastikan format anotasi sesuai dengan format yang dibutuhkan oleh YOLO (misalnya format YOLO .txt).
3. **Pembagian Dataset:** Membagi dataset yang telah diannotasi menjadi tiga bagian:
- Dataset pelatihan (misalnya 70%)
 - Dataset validasi (misalnya 15%)
 - Dataset pengujian (misalnya 15%)

Tahap 2: Pelatihan Model YOLOv8n

1. **Instalasi dan Konfigurasi:** Menginstal library ultralytics untuk menggunakan YOLOv8.
2. **Konfigurasi File YAML:** Membuat file konfigurasi YAML yang berisi informasi tentang path dataset, jumlah kelas, dan nama-nama kelas.
3. **Pelatihan Model:** Melakukan *transfer learning* dengan melatih model YOLOv8n (*pretrained weights*) menggunakan dataset yang telah diannotasi. Mahasiswa perlu menentukan hyperparameter yang sesuai (misalnya jumlah *epochs*, *batch size*, *learning rate*).
4. **Pemantauan Pelatihan:** Memantau proses pelatihan menggunakan metrik yang tersedia (misalnya *loss*, *mAP*).
5. **Evaluasi Model:** Mengevaluasi kinerja model yang telah dilatih menggunakan dataset validasi untuk melihat potensi *overfitting* atau *underfitting*.

Tahap 3: Integrasi dengan IP Camera dan Pengujian

1. **Integrasi dengan IP Camera (Opsional):**
 - Jika memungkinkan, mahasiswa mencoba mengintegrasikan model YOLOv8n dengan *stream* video dari IP camera. Ini bisa menjadi tantangan tambahan dan sangat aplikatif.
 - Jika integrasi langsung dengan IP camera sulit, mahasiswa dapat menggunakan video simulasi permukaan rel yang direkam sebelumnya.
2. **Implementasi Deteksi *Real-time* atau *Near Real-time*:** Membuat *script* Python yang dapat memproses *frame* video dari IP camera (atau video simulasi) dan melakukan deteksi kecacatan menggunakan model YOLOv8n yang telah dilatih.
3. **Pengujian Model:** Menguji kinerja model pada dataset pengujian yang belum pernah dilihat sebelumnya.
4. **Evaluasi Kinerja:** Menghitung metrik evaluasi yang relevan (misalnya *Precision*, *Recall*, *F1-score*, *mAP@0.5*, *mAP@0.5:0.95*) untuk mengukur kinerja model dalam mendeteksi berbagai jenis kecacatan.

Luaran Tugas:

Setiap kelompok harus menghasilkan:

1. Laporan Proyek:

- Pendahuluan (latar belakang, tujuan).
- Metodologi (pengumpulan data, anotasi, pelatihan model, integrasi, evaluasi).
- Hasil dan Pembahasan (hasil pelatihan, hasil pengujian, analisis kinerja model, contoh deteksi).
- Kesimpulan dan Saran.
- Lampiran (file konfigurasi, contoh anotasi, *screenshot* kode).

2. Kode Program:

- *Script* Python untuk pelatihan model.
- *Script* Python untuk deteksi kecacatan menggunakan model terlatih (dengan atau tanpa integrasi IP camera).

3. Presentasi Proyek: Mempresentasikan hasil proyek di depan kelas, mendemonstrasikan sistem yang telah dibangun.