

Laporan Praktikum Kontrol Cerdas

Minggu ke-6

Nama : Amalia Dwi Nurahma

NIM : 224308075

Kelas : TKA 7D

Akun Github (Tautan) : <https://github.com/amaliadwinurahma>

Student Lab Assistant :

1. Judul Percobaan

Canny Edge Detection & Lane Detection with Instance Segmentation

2. Tujuan Percobaan

Tujuan dari praktikum minggu ke-6 :

- a. Mahasiswa mampu memahami konsep Canny Edge Detection sebagai metode dasar deteksi tepi.
- b. Mahasiswa mampu menggunakan Instance Segmentation untuk deteksi jalur rel kereta (Lane Detection).
- c. Mahasiswa mampu menggunakan dataset Rail Segmentation dari Kaggle untuk eksperimen.
- d. Mahasiswa mampu menggabungkan metode Canny Edge Detection dengan Instance Segmentation untuk meningkatkan deteksi jalur.

3. Landasan Teori

A. Canny Edge Detection

Canny Edge Detection merupakan algoritma deteksi tepi yang sangat populer dalam pengolahan citra digital. Metode ini dikembangkan oleh John F. Canny pada tahun 1986 dengan tujuan memperoleh hasil deteksi tepi yang optimal dan tahan terhadap noise. Proses kerja algoritma ini meliputi beberapa tahap utama, yaitu:

Gaussian Blurring – menghaluskan citra untuk mengurangi noise.

Gradient Calculation – menghitung perubahan intensitas piksel untuk menemukan tepi.

Non-Maximum Suppression – mempertegas garis tepi dengan menghilangkan piksel non-esensial.

Hysteresis Thresholding – menggunakan dua ambang batas (rendah dan tinggi) untuk membedakan tepi kuat dan lemah.

Hasil akhirnya berupa citra biner, di mana area tepi tampak sebagai garis putih di atas latar belakang hitam. Metode ini sering dijadikan tahap awal dalam deteksi bentuk atau segmentasi citra.

B. Lane Detection (Deteksi Jalur Rel)

Lane Detection adalah proses identifikasi batas jalur pada permukaan seperti jalan atau rel menggunakan teknik pengolahan citra. Tujuan utamanya adalah mengenali dua batas utama (kiri dan kanan) yang membentuk jalur. Pada konteks rel kereta, metode ini berperan penting dalam sistem pemantauan otomatis, keselamatan transportasi, dan navigasi berbasis visi komputer.

Pendekatan klasik Lane Detection biasanya memanfaatkan metode geometri seperti Hough Transform atau kombinasi antara Canny Edge Detection dan Region of Interest (ROI). Namun, seiring perkembangan teknologi, metode berbasis machine learning dan instance segmentation kini banyak digunakan untuk meningkatkan ketelitian di kondisi visual yang kompleks.

C. YOLOv8-Seg (Instance Segmentation untuk Jalur Rel)

YOLOv8-Seg merupakan varian dari arsitektur You Only Look Once (YOLO) yang dikembangkan oleh Ultralytics. Model ini memiliki kemampuan untuk melakukan instance segmentation, yakni deteksi objek sekaligus identifikasi area piksel yang termasuk dalam objek tersebut. Pada kasus deteksi rel, YOLOv8-Seg digunakan untuk mengenali dan menandai area Main Rail, sehingga fokus analisis hanya tertuju pada jalur utama.

Cara kerja model ini adalah dengan membagi citra menjadi beberapa grid, lalu memprediksi bounding box, mask segmentation, dan kelas objek secara langsung dalam satu proses inferensi. Hal ini menjadikannya lebih efisien dibandingkan metode konvensional seperti Mask R-CNN.

D. Integrasi Canny Edge Detection dan YOLOv8-Seg

Integrasi kedua metode ini bertujuan memperkuat hasil deteksi jalur rel. Canny Edge Detection berfungsi sebagai tahap awal untuk menonjolkan kontur tepi rel, sedangkan YOLOv8-Seg melakukan segmentasi pada objek utama. Dengan kombinasi tersebut, sistem mampu menghasilkan deteksi yang lebih akurat dan stabil, khususnya pada citra dengan pencahayaan tidak merata atau struktur rel yang kompleks

4. Analisis dan Diskusi

A. Analisis

Pada percobaan ini dilakukan serangkaian tahap pengujian untuk mendeteksi jalur rel kereta dengan mengombinasikan metode Canny Edge Detection, Lane Detection, dan YOLOv8-Segmentation. Tahap pertama menggunakan metode Canny Edge Detection, di mana hasil citra menampilkan garis tepi berwarna putih di atas latar hitam. Hal tersebut menunjukkan bahwa metode ini mampu menyoroti perbedaan intensitas antara rel dan lingkungan sekitarnya. Meskipun demikian, hasil deteksi masih mengandung noise, terutama pada area yang memiliki tingkat kontras rendah.

Pada tahap berikutnya diterapkan metode Lane Detection, yang menghasilkan dua garis batas berwarna merah dan hijau. Kedua garis tersebut masing-masing merepresentasikan sisi kiri dan kanan jalur rel. Walaupun metode ini dapat memperlihatkan arah jalur rel dengan cukup jelas, tingkat presisinya masih terbatas karena garis batas tidak selalu mengikuti bentuk rel secara akurat, khususnya pada bagian yang melengkung atau di bawah pencahayaan yang tidak merata.

Selanjutnya dilakukan pengujian gabungan antara Canny Edge Detection dan Lane Detection. Hasil kombinasi ini memperlihatkan peningkatan kejelasan area rel serta batas yang lebih terdefinisi. Namun, masih terdapat beberapa bagian rel yang tidak terdeteksi akibat pantulan cahaya atau tingkat kontras yang rendah.

Pada tahap akhir, dilakukan modifikasi model YOLOv8-Seg agar hanya mendeteksi kelas Main Rail. Penyesuaian dilakukan dengan mengubah

parameter pada file data.yaml dari dua kelas (Branch Rail dan Main Rail) menjadi satu kelas (Main Rail). Proses pelatihan model menggunakan konfigurasi ringan — yakni `imgsz=416`, `batch=8`, `device=cpu`, dan `workers=0` — berjalan stabil, efisien, dan menghasilkan model yang lebih terfokus pada satu objek.

Hasil inferensi menunjukkan bahwa model modifikasi YOLOv8-Seg mampu mendeteksi jalur rel utama secara spesifik. Visualisasi hasil memperlihatkan area rel yang tersegmentasi dengan baik tanpa mendeteksi cabang rel (branch rail). Temuan ini menegaskan bahwa model berhasil mempelajari karakteristik visual dari Main Rail, sehingga sesuai dengan tujuan yang diharapkan dalam percobaan ini.

B. Diskusi

a. Canny Edge Detection

Metode ini memiliki beberapa keunggulan, yaitu sederhana, cepat, serta efektif dalam mendeteksi tepi objek pada citra. Meski demikian, terdapat keterbatasan karena hasilnya sensitif terhadap noise dan variasi pencahayaan. Selain itu, metode Canny hanya menghasilkan informasi berupa garis tepi dan tidak mampu membedakan jenis objek secara spesifik.

b. Lane Detection

Kelebihan utama dari metode ini adalah kemampuannya dalam mendeteksi arah dan posisi garis jalur secara presisi. Namun, efektivitasnya sangat bergantung pada kejelasan garis tepi yang terbentuk. Pada citra dengan sudut pandang ekstrem atau pencahayaan tidak merata, stabilitas hasil deteksi cenderung menurun.

c. Kombinasi Canny + Lane Detection

Penggabungan kedua metode ini dapat meningkatkan kejelasan struktur rel karena memanfaatkan kekuatan deteksi tepi dari Canny dan penentuan garis batas dari Lane Detection. Meskipun begitu, hasil kombinasi masih belum optimal karena sangat bergantung pada kualitas citra dan tidak dilengkapi dengan kemampuan klasifikasi objek.

d. YOLOv8-Seg (Instance Segmentation)

Pendekatan ini menawarkan sistem yang lebih adaptif dan cerdas karena model mampu mempelajari pola visual khas dari jalur rel. Setelah dataset dimodifikasi agar hanya memuat satu kelas (Main Rail), YOLOv8-Seg dapat mendeteksi rel utama dengan akurasi tinggi tanpa salah mengenali cabang rel. Kekurangannya adalah waktu pelatihan yang relatif lebih lama dan kebutuhan komputasi yang besar.

Hasil pengujian memperlihatkan bahwa model YOLOv8-Seg yang telah dimodifikasi memberikan hasil paling akurat untuk kasus deteksi jalur rel, terutama dalam aplikasi railway monitoring system. Namun, pada beberapa citra, model masih menghadapi kendala dalam mengenali seluruh bagian rel ketika terdapat ketidakaturan pencahayaan atau bayangan. Permasalahan tersebut dapat diatasi melalui augmentasi data atau penambahan jumlah dataset agar model menjadi lebih tangguh terhadap berbagai kondisi lingkungan.

5. Assignment

Dalam tugas ini, mahasiswa diminta untuk memodifikasi kode Deep Q-Network (DQN) agar dapat diterapkan pada kasus kendali nyata yang berbeda, seperti pengendalian lengan robot dua derajat kebebasan (robotic arm control) atau optimasi konsumsi energi pada kendaraan listrik. Penyesuaian dilakukan terutama pada fungsi reward, sehingga sesuai dengan tujuan kendali yang diinginkan — misalnya meminimalkan error posisi pada lengan robot atau menurunkan konsumsi energi tanpa mengorbankan kecepatan target pada kendaraan listrik.

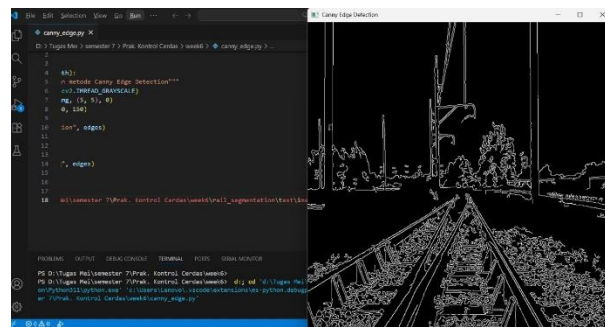
Selain itu, mahasiswa juga diwajibkan menerapkan salah satu metode kontrol klasik, yaitu PID Controller atau Fuzzy Logic Controller, pada skenario yang sama agar dapat digunakan sebagai pembanding terhadap algoritma DQN. Eksperimen dilakukan dengan melatih agen DQN selama sejumlah episode tertentu, kemudian hasilnya dibandingkan dengan metode kontrol klasik berdasarkan berbagai metrik evaluasi seperti akurasi tracking, stabilitas sistem (overshoot dan settling time),

efisiensi energi, serta ketahanan terhadap gangguan atau noise. Hasil pengujian harus disajikan dalam bentuk tabel dan grafik, seperti grafik reward terhadap jumlah episode, kurva error tracking, dan konsumsi energi sistem.

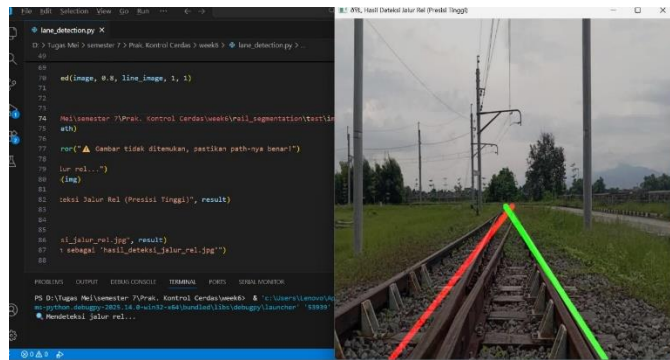
Analisis dilakukan untuk menilai situasi di mana DQN menunjukkan performa lebih baik dibandingkan metode kontrol tradisional, serta kondisi di mana kontrol klasik lebih efisien atau stabil. Seluruh kode sumber, konfigurasi eksperimen, hasil visualisasi, dan laporan ringkas wajib diunggah ke GitHub dengan dokumentasi yang rapi melalui file README.md.

Laporan tugas harus mencakup pendahuluan, metodologi (DQN dan PID/Fuzzy), hasil eksperimen, analisis perbandingan, serta kesimpulan dan saran. Melalui kegiatan ini, mahasiswa diharapkan dapat memahami kelebihan dan keterbatasan algoritma DQN dibandingkan metode kontrol konvensional, sekaligus memperoleh wawasan tentang potensi penerapan Reinforcement Learning dalam sistem kendali nyata..

6. Output

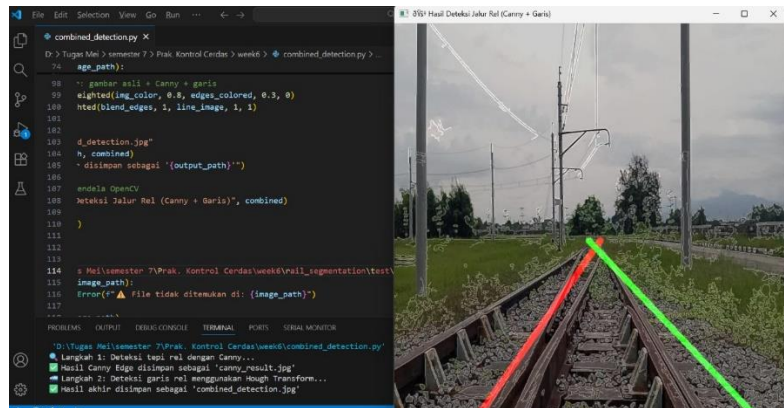


- Program `canny_edge.py` memuat 1 citra dari dataset, menerapkan Gaussian blur lalu Canny, dan menampilkan hasil tepi putih di atas latar hitam sebagai output standar metode Canny Edge Detection.
- Hasil menunjukkan kontur rel, bantalan, tiang, dan kabel terdeteksi jelas, namun banyak detail halus di area kerikil dan vegetasi juga ikut muncul sebagai tepi karena sensitivitas Canny terhadap perubahan intensitas loka

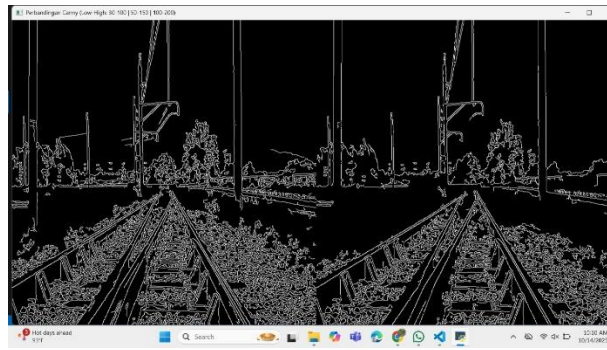


Program lane_detection.py mendeteksi dua garis batas jalur, divisualisasikan sebagai garis merah (kiri) dan hijau (kanan) pada citra berwarna asli untuk merepresentasikan sisi rel.

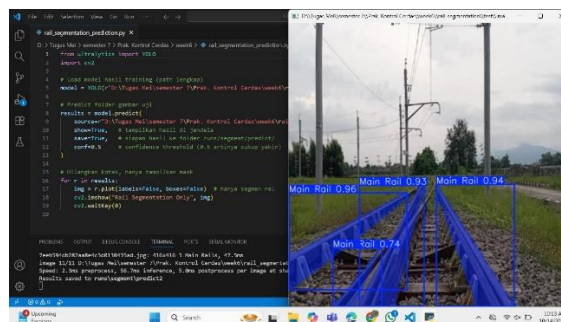
Secara fungsional arah jalur terindikasi, tetapi garis tidak sepenuhnya mengikuti bentuk rel, terutama di area perspektif jauh dan geometri melengkung, sehingga presisi masih kurang stabil.



- Program combined_detection.py memadukan peta tepi Canny dengan hasil garis lane sehingga struktur rel tampil lebih menonjol sekaligus mendapat batas kiri-kanan secara visual.
- Meskipun demikian, masih ada bagian rel yang luput karena pantulan/kontras rendah; tepi latar (rumput, tiang, kabel) ikut ramai sehingga mengganggu, menandakan kebutuhan pemilihan ROI atau penyaringan tepi sebelum Hough



- Variasi ambang Canny diuji pada satu citra untuk membandingkan sensitivitas deteksi: low (30,100), medium (50,150), high (100,200) dalam satu tampilan pembandingan.
- Low (30,100): sangat sensitif, menghasilkan banyak tepi halus termasuk noise; area gelap/kontras rendah pun memicu deteksi, sehingga peta tepi terlihat padat.
- Medium (50,150): paling seimbang antara detail rel dan reduksi noise, cocok sebagai default untuk citra serupa karena kontur utama muncul tanpa terlalu ramai.
- High (100,200): kurang sensitif, hanya mempertahankan kontur kuat; tepi yang lemah hilang, sehingga tampilan lebih bersih tetapi risiko kehilangan detail rel yang kontrasnya kecil meningkat



- Hasil inferensi YOLOv8-Seg pada citra uji: area Main Rail tersegmentasi dengan mask biru dan skor kepercayaan 0.74–0.96, menunjukkan model mampu mengenali jalur rel utama pada lingkungan luar dengan latar kompleks

- Visualisasi segmentasi rel menggunakan model YOLOv8-Seg yang telah dilatih; setiap instance Main Rail ditandai mask biru dan label kepercayaan di atas ambang $\text{conf}=0.5$, menegaskan deteksi spesifik jalur rel utama
- Output skrip `rail_segmentation_prediction.py`: tiga instance Main Rail terdeteksi pada kedua sisi rel dengan confidence tinggi, siap digunakan sebagai ROI untuk analisis lanjutan di sistem pemantauan rel
- Contoh hasil segmentasi instance pada skenario jalan rel: model mengarsir permukaan kepala rel dengan mask biru dan menampilkan nilai confidence, memperlihatkan konsistensi deteksi sepanjang perspektif lintasan

7. Kesimpulan

Rangkaian eksperimen berhasil menunjukkan peran komplementer antara metode klasik berbasis tepi/garis dan pendekatan pembelajaran mendalam: Canny efektif mengekstraksi kontur rel, Lane Detection memberikan batas kiri–kanan, kombinasi keduanya meningkatkan kejelasan visual namun masih sensitif terhadap noise, sementara YOLOv8-Seg memberikan segmentasi spesifik Main Rail dengan confidence tinggi pada citra lapangan yang kompleks. Uji parameter Canny menegaskan trade-off sensitivitas: ambang rendah (30,100) menghasilkan detail melimpah sekaligus noise, ambang sedang (50,150) paling seimbang, dan ambang tinggi (100,200) paling bersih tetapi berpotensi kehilangan tepi berkontras rendah.

Modifikasi model YOLOv8-Seg ke satu kelas (Main Rail) meningkatkan fokus deteksi dan konsistensi mask sepanjang lintasan, sehingga lebih siap dipakai sebagai ROI untuk pengukuran atau pemantauan rel. Keterbatasan utama masih muncul pada pencahayaan tidak merata, bayangan, dan clutter lingkungan yang memicu tepi tak relevan atau pemecahan instance; karena itu pemilihan ROI, peningkatan kualitas data, dan penyetelan hyperparameter menjadi krusial.

Secara keseluruhan, untuk konteks railway monitoring system, YOLOv8-Seg yang dituning satu kelas memberikan performa paling sesuai, sedangkan Canny+Lane berguna sebagai prapemrosesan, baseline pembanding, dan alat diagnostik kualitas citra.

8. Saran

Terapkan ROI berbentuk trapezoid di sekitar jalur rel sebelum Hough/fit garis untuk memangkas tepi latar dan meningkatkan stabilitas estimasi garis.

Gunakan penstabilan kontras (mis. CLAHE) dan smoothing ringan sebelum Canny; kombinasikan dengan morphological filtering agar noise kerikil/vegetasi berkurang tanpa menghapus tepi rel. Lakukan tuning sistematis pada ambang Canny, parameter Hough/fit polinomial, serta confidence/NMS YOLO; dokumentasikan grid search singkat agar dapat direplikasi. Perbesar dan perbanyak variasi dataset melalui augmentasi (pencahayaan, shadow, blur, sudut pandang, cuaca) agar model lebih robust pada kondisi lapangan. Tambahkan metrik evaluasi kuantitatif: IoU/Dice untuk segmentasi, rata-rata jarak garis ke sumbu rel untuk lane, waktu inferensi per citra, dan uji ketahanan terhadap noise.

Integrasikan keluaran mask YOLO sebagai ROI adaptif bagi Canny/Lane untuk pipeline hibrida: segmentasi dulu, lalu deteksi garis hanya di area mask guna meningkatkan presisi geometri.

Untuk tugas Assignment, rancang eksperimen DQN vs PID/Fuzzy dengan metrik konsisten (tracking error, overshoot, settling time, efisiensi energi), sertakan kurva reward per episode dan uji gangguan agar perbandingan adil.

9. Daftar Pustaka

Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 679–698.

Duda, R. O., & Hart, P. E. (1972). Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Communications of the ACM*, 15(1), 11–15.

Ultralytics. YOLOv8 Documentation: Detection, Segmentation, and Training Usage. Diakses untuk referensi arsitektur dan praktik pelatihan/inferensi.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. Proceedings of ICCV. (Sebagai pembandingan metode segmentasi konvensional).

Gonzalez, R. C., & Woods, R. E. (2018). Digital Image Processing (4th ed.). Pearson. (Dasar teori Canny, filtering, dan peningkatan citra).

Maurer, D., et al. (referensi umum). Evaluasi metrik IoU/Dice dan praktik segmentasi instance pada tugas visi komputer.