

Laporan Praktikum Kontrol Cerdas

Nama : Aureyza Pandu Qinara
NIM : 224308078
Kelas : TKA - 7D
Akun Github (Tautan) : <https://github.com/aureyzapandu>
Student Lab Assistant :-

Judul Percobaan

Week 6: *Canny Edge Detection & Lane Detection with Instance Segmentation*

1. Tujuan Percobaan

Tujuan dari praktikum “*Canny Edge Detection & Lane Detection with Instance Segmentation*”, mahasiswa diharapkan mampu:

- a) Memahami konsep *Canny Edge Detection* sebagai metode dasar deteksi tepi.
- b) Menggunakan *Instance Segmentation* untuk deteksi jalur rel kereta (*Lane Detection*).
- c) Menggunakan *dataset Rail Segmentation* dari *Kaggle* untuk eksperimen.
- d) Menggabungkan metode *Canny Edge Detection* dengan *Instance Segmentation* untuk meningkatkan deteksi jalur.

2. Landasan Teori

- *Canny Edge Detection*

Canny Edge Detection merupakan metode deteksi tepi yang digunakan untuk mengidentifikasi batas objek dalam gambar berdasarkan perbedaan intensitas piksel (Canny, 1986). Proses deteksi tepi ini terdiri dari beberapa tahapan utama:

- a. ***Gaussian Blur***: Digunakan untuk mengurangi *noise* pada gambar sebelum proses deteksi tepi dimulai. Hal ini penting karena *noise* dapat menghasilkan deteksi yang tidak akurat.
- b. ***Gradient Calculation***: Setelah proses blur, gradien citra dihitung untuk menemukan area dengan perubahan intensitas yang signifikan.

- c. ***Non-Maximum Suppression (NMS)***: Tujuannya adalah untuk menipiskan hasil gradien dan menjaga hanya titik tepi yang paling signifikan.
- d. ***Hysteresis Thresholding***: Menentukan tepi dengan menggunakan dua ambang batas, yaitu tinggi dan rendah. Tepi yang memiliki nilai lebih tinggi dari ambang batas tinggi dianggap tepi yang kuat, sementara yang di bawah ambang batas rendah dianggap bukan tepi.

Canny Edge Detection dikenal karena ketahanannya terhadap *noise* dan kemampuannya untuk mendeteksi tepi yang jelas, meskipun kadang kurang efektif dalam mendeteksi tepi yang lebih halus atau terdistorsi oleh *noise*.

- ***Lane Detection dengan Instance Segmentation***

Lane Detection adalah proses deteksi jalur atau garis jalan, yang memiliki aplikasi penting dalam sistem kendaraan otonom dan navigasi. Salah satu metode modern untuk deteksi jalur adalah *Instance Segmentation*, yang mengkombinasikan teknik deteksi objek dan segmentasi piksel untuk membedakan antara objek yang terpisah dalam gambar (Redmon et al., 2016). Berbeda dengan metode berbasis tepi seperti *Canny*, *Instance Segmentation* mampu memberikan informasi lebih terperinci dengan membedakan objek secara individual.

Metode ini menggunakan model YOLOv8-seg, sebuah model *deep learning* untuk segmentasi objek instance. YOLOv8-seg memiliki kemampuan untuk mendeteksi jalur rel kereta dengan akurat, mengidentifikasi objek dalam gambar dan menandai posisi serta bentuk objek tersebut. Dengan menggunakan algoritma ini, jalur rel dapat terdeteksi meskipun lingkungan sekitar tampak kompleks atau terdistorsi.

- ***Gabungan Canny Edge Detection dengan Instance Segmentation***

Menggabungkan *Canny Edge Detection* dengan *Instance Segmentation* memungkinkan peningkatan akurasi deteksi jalur rel. *Canny Edge Detection* berfungsi untuk mendeteksi tepi-tepi penting pada gambar, sementara *Instance Segmentation* memberikan segmentasi yang lebih terperinci, memungkinkan sistem untuk membedakan jalur rel dari elemen lain yang ada di sekitarnya. Dengan menggabungkan kedua metode ini, deteksi jalur

menjadi lebih kuat, karena dapat mengurangi kesalahan deteksi yang sering terjadi hanya menggunakan satu metode saja (Cheng et al., 2023).

3. Analisis dan Diskusi

Pada eksperimen ini, digunakan dua metode utama untuk mendeteksi jalur rel, yaitu *Canny Edge Detection* dan *Instance Segmentation* (YOLOv8-seg). *Canny Edge Detection* berfungsi untuk menyorot tepi atau kontur pada gambar berdasarkan perubahan intensitas piksel, sementara YOLOv8-seg digunakan untuk melakukan segmentasi instance secara mendalam terhadap objek jalur rel.

Dari hasil pengujian:

- *Canny Edge Detection* mampu menampilkan garis tepi rel dengan baik pada gambar yang memiliki kontras tinggi dan pencahayaan merata. Namun, metode ini sensitif terhadap *noise* dan bayangan, sehingga beberapa bagian rel bisa terputus atau muncul garis palsu di area non-rel.
- *Instance Segmentation* (YOLOv8-seg) memberikan hasil yang jauh lebih stabil dan akurat, karena model ini memanfaatkan pembelajaran mendalam untuk mengenali bentuk dan pola rel meskipun kondisi pencahayaan atau latar belakang berubah.
- Kombinasi kedua metode menghasilkan deteksi yang lebih informatif. Hasil segmentasi dari YOLO memperlihatkan area jalur secara menyeluruh, sementara hasil *Canny* menambahkan detail tepi yang memperjelas batas antar komponen rel. Kombinasi ini meningkatkan kejelasan visual serta membantu verifikasi bentuk jalur.

Canny lebih cocok digunakan pada sistem ringan atau perangkat dengan keterbatasan komputasi, karena tidak memerlukan model besar dan berjalan cepat secara *real-time*. Namun, metode ini kurang akurat jika kondisi lingkungan kompleks atau kontras rendah.

Akurasi YOLOv8-seg dapat ditingkatkan dengan:

- Menyesuaikan *confidence threshold* dan *IoU threshold* untuk mengurangi *false detection*.
- Melatih ulang model menggunakan *dataset* rel lokal agar lebih spesifik terhadap karakteristik rel Indonesia.
- Menggunakan model yang lebih besar (misal yolov8m-seg atau yolov8l-seg) untuk peningkatan detail segmentasi.

Sistem gabungan ini bisa menjadi komponen penting dalam perception module kendaraan rel otomatis. *Canny Edge* dapat berfungsi untuk *real-time edge tracking*, sementara YOLOv8-seg dapat mendeteksi dan mengidentifikasi area jalur yang valid.



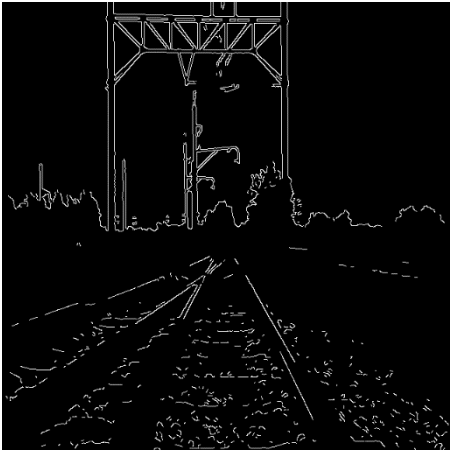
4. Assignment

1. Melakukan studi literatur tentang *Canny Edge Detection* dan *Instance Segmentation* untuk Deteksi Jalur Rel.
2. Membuat akun dan repository GitHub untuk version control proyek.
3. Menginstal Python, VSCode, Git, dan *library* yang dibutuhkan.
4. Menyiapkan *environment* dan mengunduh *dataset* Rail Segmentation, dari Kaggle dan rail-dataset.pt
5. Mengimplementasikan *Canny Edge Detection* untuk mendeteksi tepi pada gambar
6. Mengimplementasikan *Lane Detection* dengan YOLOv8 *Instance Segmentation* untuk mendeteksi jalur rel
7. Ubah parameter threshold pada *Canny Edge Detection* dan lihat bagaimana perubahan tersebut mempengaruhi hasil deteksi tepi.
8. Sesuaikan model YOLOv8-seg (rail-dataset.pt) agar hanya mendeteksi jalur rel dengan menyaring hasil deteksi untuk objek rel saja.
9. Gabungkan hasil deteksi dari *Canny Edge Detection* dan YOLOv8 untuk meningkatkan akurasi dalam mendeteksi jalur rel pada gambar.
10. Evaluasi hasil deteksi jalur rel menggunakan kedua metode (*Canny* dan YOLOv8-seg) serta gabungan keduanya untuk menentukan akurasi deteksi.
11. •

12. Simpan semua hasil eksperimen, termasuk gambar deteksi, dan model yang digunakan.

13. Melakukan commit dan push ke GitHub sebagai dokumentasi akhir

5. Data dan Output Hasil Pengamatan

No	Variabel	Hasil Pengamatan
1	Hasil Modifikasi model <i>dataset</i> YOLOv8-seg (<i>rail-dataset.pt</i>)	
2	Hasil Pengujian <i>Canny Edge Detection</i> Parameter <i>Threshold 1</i> : (50, 150)	
3	Hasil Pengujian <i>Canny Edge Detection</i> Parameter <i>Threshold 2</i> : (200, 250)	

6. Kesimpulan

- a) YOLOv8 *Instance Segmentation* efektif dalam mendeteksi jalur rel dengan membedakan objek (*Main Rail* dan *Branch Rail*) secara akurat, memberikan hasil deteksi yang lebih terperinci dengan probabilitas yang jelas.
- b) *Canny Edge Detection* dengan *threshold* rendah (50, 150) menghasilkan deteksi tepi yang lebih kasar dan banyak *noise*, sementara *threshold* tinggi (200, 250) memberikan hasil yang lebih tajam dan lebih fokus pada tepi yang signifikan.
- c) Penggabungan kedua metode (*Canny Edge Detection* dan YOLOv8) dapat meningkatkan akurasi deteksi jalur rel dengan memanfaatkan kekuatan masing-masing metode—*Canny* untuk mendeteksi tepi dan YOLO untuk segmentasi objek secara lebih akurat.

7. Saran

- a) Menyesuaikan *threshold* lebih lanjut agar mendapatkan hasil tepi yang lebih halus dan lebih sensitif terhadap perubahan kecil pada jalur rel.
- b) Modifikasi model YOLOv8 dengan *fine-tuning* pada dataset spesifik agar dapat mendeteksi jalur rel dengan lebih tepat.
- c) Uji algoritma pada gambar dengan variasi kondisi pencahayaan dan *noise* untuk memastikan keandalan metode di berbagai kondisi lapangan.

8. Daftar Pustaka

- Canny, J., 1986. A Computational Approach to Edge Detection. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8, 679–698.
<https://doi.org/10.1109/TPAMI.1986.4767851>
- Cheng, W., Wang, X., Mao, B., 2023. Research on Lane Line Detection Algorithm Based on Instance Segmentation. Sensors 23, 789.
<https://doi.org/10.3390/s23020789>
- Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection.
<https://doi.org/10.48550/arXiv.1506.02640>