

Laporan Praktikum Minggu Ke-6 Kontrol Cerdas

Minggu ke-6

Nama : Fahrul Ikhza Fadilla
NIM : 224308055
Kelas : TKA 7C
Akun Github (Tautan) : <https://github.com/FahrulIkhza>

1. Judul Percobaan

Canny Edge Detection & Lane Detection with Instance Segmentation

2. Tujuan Percobaan

Tujuan dari praktikum “Human Pose Estimation dengan YOLOv8”, mahasiswa diharapkan mampu:

- Memahami konsep Canny Edge Detection sebagai metode dasar deteksi tepi.
- Menggunakan Instance Segmentation untuk deteksi jalur rel kereta (Lane Detection).
- Menggunakan dataset Rail Segmentation dari kagle untuk eksperimen.
- 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 dikembangkan oleh John F. Canny pada tahun 1986. Metode ini digunakan untuk menemukan batas atau tepi objek pada gambar dengan hasil yang akurat dan minim kesalahan. Prosesnya terdiri dari empat tahap utama, yaitu penghalusan menggunakan filter Gaussian untuk mengurangi noise, perhitungan gradien intensitas untuk mengetahui perubahan nilai piksel, non-maximum suppression untuk menyisakan tepi utama, serta hysteresis thresholding yang berfungsi menghubungkan tepi lemah dan tepi kuat. Melalui tahapan ini, Canny Edge Detection dapat menghasilkan deteksi tepi yang jelas dan halus sehingga sering digunakan dalam berbagai aplikasi seperti segmentasi objek dan pengenalan pola citra.

b. Instance Segmentation

Instance Segmentation merupakan teknik dalam bidang computer vision yang mampu mengenali dan memisahkan setiap objek individual dalam suatu gambar. Berbeda dengan semantic segmentation yang hanya mengelompokkan jenis objek, instance segmentation mampu membedakan setiap objek meskipun berasal dari kelas yang sama. Teknik ini memberikan mask atau batas area untuk setiap objek sehingga sistem dapat memahami bentuk dan letak objek secara lebih detail. Model populer seperti Mask R-CNN dan YOLOv8-seg banyak digunakan karena mampu memberikan hasil segmentasi yang akurat dan efisien untuk berbagai aplikasi seperti deteksi kendaraan, pengawasan lalu lintas, serta navigasi otonom.

c. Rail Segmentation

Rail Segmentation adalah metode dalam computer vision yang digunakan untuk mengenali dan mendeteksi jalur rel kereta api dari citra atau video. Teknik ini penting untuk sistem pemantauan infrastruktur perkeretaapian karena mampu membantu mendeteksi kerusakan atau gangguan pada rel. Dalam penerapannya, rail segmentation biasanya menggunakan algoritma deep learning seperti CNN, YOLOv8-seg, atau DeepLabV3+ untuk mengenali bentuk jalur rel secara akurat bahkan dalam kondisi pencahayaan atau lingkungan yang bervariasi. Dengan metode ini, inspeksi rel dapat dilakukan secara otomatis dan efisien tanpa harus dilakukan pemeriksaan manual.

d. YOLOv8-seg

YOLOv8-seg merupakan pengembangan dari model YOLOv8 yang dirancang khusus untuk melakukan instance segmentation. Model ini tidak hanya mampu mendeteksi keberadaan objek, tetapi juga memberikan batas area atau mask pada setiap objek yang terdeteksi. Keunggulan YOLOv8-seg terletak pada kecepatan pemrosesan dan akurasi yang lebih tinggi dibandingkan versi sebelumnya. Arsitekturnya yang efisien menjadikannya cocok digunakan pada aplikasi real-time seperti deteksi jalur rel, analisis video, maupun sistem navigasi kendaraan otonom berbasis kecerdasan buatan.

e. Konversi Video ke Gambar

Konversi video ke gambar merupakan proses mengubah setiap frame dari video menjadi citra tunggal. Langkah ini sering digunakan dalam pembuatan dataset untuk pelatihan model machine learning serta analisis pergerakan objek. Proses konversi dapat dilakukan dengan bantuan pustaka seperti OpenCV, FFmpeg, atau ImageIO. Dalam konteks deteksi objek dan segmentasi, proses ini sangat membantu karena memungkinkan pengambilan data dari video menjadi kumpulan gambar yang dapat digunakan untuk melatih model secara lebih efisien.

f. Roboflow

Roboflow adalah platform yang menyediakan layanan untuk mengelola dan melatih dataset dalam proyek computer vision. Melalui platform ini, pengguna dapat mengunggah gambar, melakukan preprocessing, dan mengonversi dataset ke berbagai format yang sesuai dengan model seperti YOLO, TensorFlow, atau PyTorch. Selain itu, Roboflow juga menyediakan fitur AutoML dan pelatihan berbasis cloud sehingga proses pelatihan model dapat dilakukan tanpa perangkat keras tambahan. Dengan fitur yang lengkap, Roboflow menjadi alat yang sangat membantu dalam pengembangan model deteksi dan segmentasi berbasis deep learning.

4. Analisis dan Diskusi

Analisis Hasil:

a. Seberapa Baik Deteksi Jalur dengan Instance Segmentation dibandingkan Canny?

Metode **Instance Segmentation**, seperti yang digunakan pada model **YOLOv8-seg**, memiliki kemampuan yang lebih baik dibandingkan **Canny Edge Detection** dalam mendeteksi jalur rel. Hal ini karena Instance Segmentation dapat mengenali dan membedakan setiap objek di dalam gambar secara spesifik, sehingga hasil segmentasinya lebih presisi dan akurat.

Sementara itu, **Canny Edge Detection** hanya berfungsi mendeteksi tepi berdasarkan perubahan intensitas tanpa memahami konteks dari objek yang terdeteksi. Meski demikian, keunggulan utama Canny adalah prosesnya yang **lebih cepat dan ringan**, sehingga cocok untuk aplikasi **real-time** atau perangkat dengan kemampuan komputasi terbatas. Dengan kata lain, YOLOv8-seg unggul dari sisi akurasi dan pemahaman visual, sedangkan Canny lebih unggul dari sisi kecepatan dan efisiensi.

b. Apakah Kombinasi Kedua Metode Dapat Meningkatkan Akurasi?

Kombinasi kedua metode ini dapat memberikan hasil deteksi yang lebih baik. **Canny Edge Detection** dapat digunakan sebagai langkah **pra-pemrosesan (pre-processing)** sebelum gambar dimasukkan ke model **YOLOv8-seg**. Dengan cara ini, fitur tepi jalur dapat ditonjolkan lebih jelas, sementara noise atau gangguan dapat dikurangi. Hasilnya, model deep learning seperti YOLOv8-seg bisa lebih fokus pada bagian penting dari gambar yang berhubungan langsung dengan jalur rel. Selain itu, hasil deteksi dari kedua metode juga dapat saling divalidasi — misalnya, dengan membandingkan hasil segmentasi dari YOLOv8-seg dengan tepi yang dideteksi oleh Canny untuk memastikan tidak ada kesalahan atau area jalur yang terlewat. Pendekatan ini membuat sistem deteksi menjadi lebih akurat dan andal.

c. **Apa Dampak Perubahan Parameter Canny (Thresholds) terhadap Hasil Deteksi?**

Dalam metode **Canny Edge Detection**, terdapat dua parameter penting yaitu **lower threshold** dan **upper threshold**.

- Jika nilai gradien piksel lebih kecil dari *lower threshold*, maka piksel tersebut tidak dianggap sebagai bagian dari tepi.
- Jika nilai gradien lebih besar dari *upper threshold*, piksel tersebut dikategorikan sebagai tepi kuat.

Pengaturan nilai threshold ini sangat berpengaruh terhadap hasil deteksi. Jika nilai ambang terlalu rendah, sistem akan mendeteksi terlalu banyak tepi, termasuk noise yang tidak diperlukan. Sebaliknya, jika ambang terlalu tinggi, banyak tepi penting justru tidak terdeteksi sehingga jalur rel bisa tampak terputus atau hilang. Oleh karena itu, pemilihan parameter threshold yang tepat perlu disesuaikan dengan **kondisi pencahayaan, tingkat kontras, dan kualitas gambar** agar hasil deteksi tepi tetap akurat dan stabil.

Diskusi :

a. **Kapan Lebih Baik Menggunakan Canny Edge Detection dibanding Instance Segmentation?**

Metode **Canny Edge Detection** lebih cocok digunakan ketika sistem memiliki keterbatasan sumber daya komputasi, seperti pada perangkat dengan prosesor standar atau tanpa dukungan GPU. Karena prosesnya ringan dan cepat, Canny sangat efisien untuk aplikasi **real-time** yang tidak membutuhkan analisis objek secara mendalam. Selain itu, metode ini juga bekerja sangat baik pada citra dengan **kontras tinggi dan pencahayaan yang stabil**, di mana tepi objek dapat terdeteksi dengan jelas tanpa gangguan noise. Canny juga menjadi pilihan ideal ketika tujuan utama hanya untuk **mendeteksi garis atau batas objek**, misalnya dalam proses pengolahan gambar sederhana tanpa perlu memahami konteks keseluruhan dari objek yang ada.

b. **Bagaimana Cara Meningkatkan Deteksi Jalur dengan Tuning Parameter YOLOv8-seg?**

Agar performa model **YOLOv8-seg** semakin optimal, diperlukan beberapa langkah penyesuaian (tuning) terhadap data dan parameter model. Pertama, gunakan **dataset berkualitas tinggi** yang memiliki anotasi (penandaan objek) akurat serta mencakup berbagai kondisi pencahayaan, sudut pandang, dan lingkungan agar model dapat belajar secara menyeluruh. Kedua, lakukan **augmentasi data** seperti pengaturan tingkat kecerahan (*brightness adjustment*), rotasi gambar, atau penambahan efek blur untuk meningkatkan kemampuan model beradaptasi terhadap kondisi nyata yang

beragam. Ketiga, terapkan proses **fine-tuning** dengan metode *transfer learning* menggunakan dataset khusus seperti *rail segmentation*. Langkah ini memungkinkan model YOLOv8-seg menyesuaikan bobotnya terhadap karakteristik jalur rel, sehingga hasil deteksinya menjadi lebih akurat dan stabil.

c. Bagaimana Metode Ini Dapat Diterapkan dalam Sistem Navigasi Kereta Otomatis?

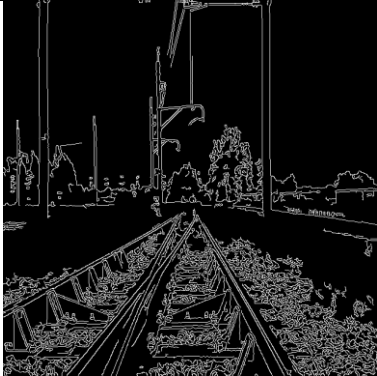
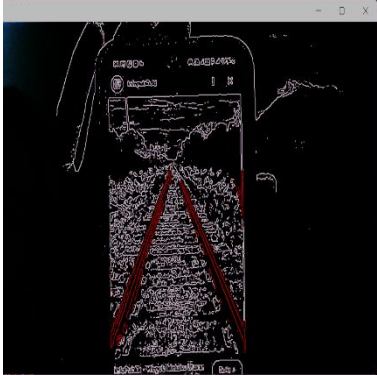
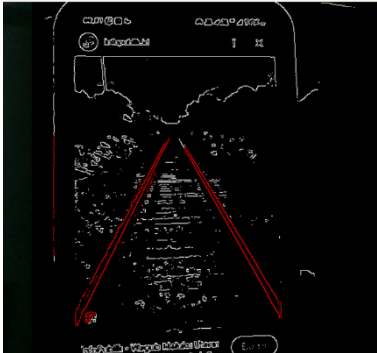
Kombinasi metode **Instance Segmentation** dan **Canny Edge Detection** dapat diterapkan secara efektif dalam sistem navigasi kereta otomatis. Instance Segmentation, seperti YOLOv8-seg, digunakan untuk **mengenali dan menandai jalur rel secara real-time**, sehingga sistem dapat mengetahui posisi lintasan dengan presisi tinggi. Selain itu, model ini dapat memberikan **peringatan dini terhadap hambatan** seperti batu, kendaraan, atau manusia yang berada di atas rel, sehingga membantu mencegah potensi kecelakaan. Informasi hasil segmentasi juga dapat dikombinasikan dengan data dari **sensor tambahan** seperti LiDAR dan IMU untuk menentukan arah serta posisi kereta secara otomatis. Lebih lanjut, metode ini dapat mendukung **maintenance prediktif**, yaitu dengan menganalisis kondisi jalur rel dari hasil segmentasi untuk mendeteksi kerusakan sejak dini sebelum menyebabkan gangguan atau kegagalan sistem.

5. Assignment

Ubah parameter Canny Edge Detection dan bandingkan hasilnya. Parameter Threshold 1: (50, 150) o Deteksi tepi lebih sensitif terhadap perubahan warna dan tekstur. o Lebih banyak noise terdeteksi, termasuk bagian yang tidak relevan seperti bayangan atau kerikil di sekitar rel. o Rel tetap terlihat, tetapi ada banyak gangguan yang bisa menghambat segmentasi. Parameter Threshold 2: (200, 250) o Hanya tepi yang sangat jelas yang terdeteksi. o Noise berkurang secara signifikan, tetapi beberapa bagian rel mungkin hilang dari deteksi. o Jika pencahayaan kurang baik, jalur rel bisa saja tidak terdeteksi dengan sempurna.

Modifikasi model YOLOv8-seg agar hanya mendeteksi jalur rel. Percobaan membuat dataset baru rail-dataset.pt untuk mendeteksi jalur rel, dengan langkah langkah berikut: Pengambilan Data: o Menggunakan video sebagai sumber dataset dan mengekstrak 1150 frame. o Frame diambil secara berkala untuk mencakup variasi sudut dan pencahayaan. Anotasi Dataset: o Menggunakan Roboflow untuk memberi label jalur rel dalam setiap frame. o Memastikan bahwa dataset memiliki beragam kondisi (siang/malam, basah/kering, dan berbagai perspektif). Upload hasil eksperimen ke GitHub dan buat laporan analisis hasil. Hasil eksperimen telah diunggah ke GitHub berserta hasil pengamatan, analisis dan diskusi beserta laporan praktikum..

6. Data dan Output Hasil Pengamatan

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

7. Kesimpulan

Dari percobaan yang telah dilakukan dapat disimpulkan bahwa:

1. Metode Canny Edge Detection efektif dalam mengekstraksi tepi dengan presisi tinggi, kontras yang jelas, lingkungan yang minim noise pada gambar dan cocok untuk sistem deteksi secara Real-time dengan cepat.
2. Metode Canny Edge Detection memerlukan turning parameter yang optimal agar tidak kehilangan detailnya atau mendeteksi tepi palsu.
3. Instance Segmentation memiliki keunggulan dalam mendeteksi dan memisahkan elemen individual dalam gambar, termasuk jalur rel dan benda

di sekitarnya, dan metode ini juga memberikan hasil deteksi yang lebih stabil dan akurat dalam kondisi lingkungan yang kompleks.

4. Metode Instance Segmentation membutuhkan dataset berlabel dan komputasi yang lebih intensif dan model yang kompleks.
5. Kombinasi Canny Edge Detection dan Instance Segmentation dapat meningkatkan akurasi deteksi jalur jalan, di mana edge detection membantu memperkuat identifikasi tepi, sementara instance segmentation memberikan pemahaman struktural.

8. Saran

Untuk meningkatkan efektivitas Canny Edge Detection, salah satu solusi yang dapat diterapkan adalah pengembangan algoritma adaptif yang secara otomatis menyesuaikan parameter threshold berdasarkan karakteristik gambar, seperti kontras dan tingkat noise. Pendekatan berbasis machine learning dapat digunakan untuk memprediksi nilai threshold optimal, mengurangi ketergantungan pada tuning manual. Selain itu, integrasi dengan filter berbasis deep learning (seperti Denoising Autoencoders) sebelum proses edge detection dapat membantu mengurangi noise tanpa menghilangkan detail tepi yang penting. Untuk aplikasi real-time, optimasi komputasi melalui paralelisasi GPU atau penggunaan library seperti OpenCV dengan optimasi CUDA dapat mempercepat pemrosesan. Sementara itu, tantangan dalam Instance Segmentation, seperti kebutuhan dataset besar dan komputasi intensif, dapat diatasi dengan teknik transfer learning. Model pre-trained dapat diadaptasi untuk deteksi jalur jalan dengan fine-tuning pada dataset yang lebih kecil, mengurangi kebutuhan sumber daya. Untuk mempercepat inferensi, penggunaan model yang lebih ringan seperti EfficientNet-based architectures atau kuantisasi model dapat diterapkan. Selain itu, augmentasi data sintetis yang dapat membantu menambah variasi dataset tanpa biaya anotasi manual yang tinggi. Untuk integrasi dengan Canny Edge Detection, pendekatan hybrid dapat digunakan, di mana instance segmentation memberikan segmentasi kasar, sementara edge detection mempertajam batas objek, sehingga meningkatkan akurasi tanpa beban komputasi berlebih. Dengan strategi ini, kedua metode dapat saling melengkapi, menghasilkan sistem deteksi yang lebih robust dan efisien.

9. Daftar Pustaka

- Gamma Akbar Sya'bana. (2021). Inspeksi Jalur Rel Kereta Api Menggunakan Kamera RGB Drone (Undergraduate Thesis). Institut Teknologi Sepuluh Nopember, Surabaya.
- Ismail, M., Kalyan, M. V. S. P., Reddy, D. R. K., Reddy, P. Y. K., Gautam, P. T. S., & B., M. I. (2020). Design and development of automated lane detection

using improved Canny edge detection method. *Psychology and Education*, 57(9), 1350-1358.

- Javeed, M. A., Ghaffar, M. A., Ashraf, M. A., Zubair, N., Metwally, A. S. M., Tag-Eldin, E. M., Bocchetta, P., Javed, M. S., & Jiang, X. (2023). Lane Line Detection and Object Scene Segmentation Using Otsu Road Thresholding and the Fast Hough Transform for Intelligent Vehicles in Complex Conditions. *Electronics*, <https://doi.org/10.3390/electronics12051079> 12(5), 1079.
- Taqy, M. A., Yahya, C. A., Hafizh, M. A., Nurakmalia, S., & Rosyani, P. (2024). Implementasi dan Analisis Metode Deteksi Tepi Canny Menggunakan OpenCV. 1(4).
- Wulanningrum, R., Handayani, A. N., & Wibawa, A. P. (2024). Perbandingan Instance Segmentation Image Pada Yolo8. *Jurnal Teknologi Informasi dan Ilmu Komputer*, <https://doi.org/10.25126/jtiik.1148288> 11(4), 753–760.
- Lugaresi, C., Tang, J., Nash, H., McClanahan, C., et al. (2019). MediaPipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*.