Laporan Praktikum Kontrol Cerdas

Week 6

Nama : Vivi Aulia Husna Wahidah

NIM : 224308023

Kelas :TKA-6A

Akun Github (Tautan) : https://github.com/vivi-aulia

Student Lab Assistant : Muhammad Mahirul Faiq

# Judul Percobaan

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

# Tujuan Percobaan

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

1. Memahami konsep Canny Edge Detection sebagai metode dasar deteksi tepi.
2. Menggunakan Instance Segmentation untuk deteksi jalur rel kereta (Lane Detection).
3. Menggunakan dataset Rail Segmentation dari Kaggle untuk eksperimen.
4. Menggabungkan metode Canny Edge Detection dengan Instance Segmentation untuk meningkatkan deteksi jalur.

# Landasan Teori

1. **Canny Edge Detection**   
   Canny Edge Detection adalah metode deteksi tepi yang dikembangkan oleh John Canny pada tahun 1986, yang bertujuan untuk memperoleh hasil deteksi tepi yang optimal dengan meminimalkan gangguan dari noise serta kesalahan deteksi. Proses dalam metode ini terdiri atas beberapa tahapan, yaitu penghalusan citra dengan filter Gaussian, perhitungan arah dan besar gradien intensitas, penipisan tepi melalui non-maximum suppression, dan penentuan tepi akhir menggunakan hysteresis thresholding. Metode ini banyak diaplikasikan dalam pemrosesan citra digital, khususnya untuk segmentasi objek dan ekstraksi fitur penting dari gambar.
2. **Instance Segmentation**  
   Instance segmentation merupakan pendekatan dalam bidang visi komputer yang menggabungkan kemampuan deteksi objek dan segmentasi citra. Teknik ini tidak hanya membedakan antar kelas objek dalam sebuah gambar, tetapi juga memisahkan masing-masing objek meskipun berasal dari kelas yang sama. Setiap objek diberikan mask tersendiri sehingga bentuk dan batasnya dapat dikenali secara individual. Model yang umum digunakan dalam instance segmentation antara lain Mask R-CNN dan YOLOv8-seg, karena mampu menghasilkan hasil deteksi yang presisi serta segmentasi yang akurat.
3. **Rail Segmentation**

Rail segmentation adalah proses pendeteksian rel kereta api dalam citra atau video, yang bertujuan untuk mendukung sistem monitoring dan perawatan infrastruktur perkeretaapian. Proses ini penting untuk mendeteksi potensi kerusakan atau penyimpangan pada rel secara otomatis. Teknologi ini biasanya mengandalkan pendekatan deep learning seperti Convolutional Neural Network (CNN), serta model segmentasi instance seperti YOLOv8-seg dan DeepLabV3+. Dengan pemrosesan visual yang canggih, rail segmentation memungkinkan pengawasan jalur kereta secara lebih efisien dan akurat.

1. **YOLOv8-seg**

YOLOv8-seg merupakan pengembangan terbaru dari arsitektur YOLO (You Only Look Once) yang telah dilengkapi dengan fitur instance segmentation. Model ini memungkinkan pendeteksian objek bersamaan dengan pembuatan mask segmentasi untuk setiap objek secara real-time. Dengan arsitektur yang efisien serta post-processing yang dioptimalkan, YOLOv8-seg memberikan kinerja yang lebih baik dalam hal kecepatan dan akurasi dibandingkan versi sebelumnya. Kemampuan ini membuat YOLOv8-seg sangat cocok diterapkan pada berbagai aplikasi vision, termasuk rail segmentation.

1. **Konversi Video ke Gambar**

Konversi video ke gambar adalah proses yang digunakan untuk mengekstraksi frame individual dari sebuah video. Proses ini sangat penting dalam pembuatan dataset untuk pelatihan model machine learning berbasis citra, seperti deteksi objek maupun segmentasi. Frame-frame hasil konversi dapat digunakan sebagai data latih atau data uji, tergantung pada kebutuhan penelitian. Beberapa pustaka populer yang digunakan dalam proses ini antara lain OpenCV, FFmpeg, dan ImageIO.

1. **Roboflow**

Roboflow adalah platform yang dirancang untuk mendukung berbagai kebutuhan dalam pengelolaan proyek computer vision. Platform ini memungkinkan pengguna untuk mengunggah dataset citra, melakukan anotasi, menerapkan augmentasi, serta mengonversi format dataset agar kompatibel dengan berbagai framework deep learning seperti YOLO, TensorFlow, dan PyTorch. Roboflow juga menyediakan fitur pelatihan model berbasis cloud, sehingga sangat membantu dalam pengembangan model AI dengan efisiensi tinggi dan kemudahan integrasi.

# Analisis dan Diskusi

**Analisis Hasil:**

1. **Penerapan Canny Edge Detection**  
   Metode Canny Edge Detection diaplikasikan pada citra rel kereta untuk mengidentifikasi tepi atau kontur objek. Dari hasil penerapan, terlihat bahwa algoritma ini mampu menghasilkan garis tepi yang cukup halus dan tajam setelah melalui tahap smoothing dengan Gaussian filter. Penggunaan non-maximum suppression turut menyaring tepi agar tidak terjadi penumpukan garis yang tidak relevan. Namun, pada citra dengan pencahayaan rendah atau banyak gangguan visual (seperti bayangan atau kerikil), algoritma ini menghasilkan garis tepi yang kurang konsisten. Oleh karena itu, metode ini dinilai belum cukup untuk digunakan secara mandiri dalam mendeteksi rel kereta.
2. **Hasil Instance Segmentation**

Model YOLOv8-seg digunakan dalam penelitian untuk melakukan instance segmentation terhadap rel kereta api. Berdasarkan hasil pengujian, model mampu membedakan rel dari objek-objek lain seperti bantalan dan batu balast. Setiap instance rel berhasil diberikan segmentasi mask yang jelas, bahkan pada kondisi gambar yang kompleks. Proses segmentasi berjalan efisien dan dapat memproses data dalam waktu singkat. Hal ini menunjukkan bahwa model sudah berjalan sesuai dengan fungsi utamanya untuk membedakan dan memisahkan objek satu per satu dalam gambar.

1. **Rail Segmentation dengan YOLOv8-seg**

YOLOv8-seg memberikan hasil segmentasi yang akurat dalam pendeteksian jalur rel. Proses pelatihan model menggunakan dataset hasil konversi video ke gambar menunjukkan performa yang stabil, di mana model mampu mengenali rel dalam berbagai sudut dan kondisi pencahayaan. Mask segmentasi yang dihasilkan mengikuti bentuk fisik rel secara presisi. Kecepatan inferensi model juga sangat baik sehingga memungkinkan penerapan secara real-time di lingkungan kerja atau sistem monitoring.

1. **Proses Konversi Video ke Gambar**

Tahapan konversi video ke gambar dilakukan untuk memperoleh dataset yang bervariasi dari satu sumber video pengamatan rel. Proses ini menghasilkan ratusan hingga ribuan frame yang siap digunakan dalam pelabelan. Data yang dihasilkan memiliki berbagai variasi posisi, tekstur, dan pencahayaan, sehingga mendukung pelatihan model agar lebih general terhadap kondisi nyata.

1. **Penggunaan Roboflow**

Roboflow digunakan untuk proses anotasi dan augmentasi dataset. Melalui platform ini, anotasi rel kereta dapat dilakukan secara presisi dengan tool labeling yang mendukung instance segmentation. Selain itu, proses augmentasi data seperti flip, brightness adjustment, dan rotation turut meningkatkan performa model. Roboflow juga mempermudah ekspor dataset ke format YOLO yang dibutuhkan oleh model YOLOv8-seg.

**Diskusi:**

Penerapan Canny Edge Detection pada tahap awal memberikan gambaran awal terhadap kontur objek, namun tidak cukup kuat digunakan untuk segmentasi rel karena keterbatasan dalam membedakan objek yang serupa secara tekstur. Oleh karena itu, metode ini hanya efektif jika digabungkan dengan teknik lanjutan berbasis deep learning.

Hasil instance segmentation dengan YOLOv8-seg memperlihatkan akurasi yang sangat baik. Model ini tidak hanya mampu mendeteksi keberadaan rel, namun juga membedakannya secara individu dari objek lain di sekitarnya. Ini sangat penting dalam sistem inspeksi otomatis pada infrastruktur perkeretaapian, karena memungkinkan pendeteksian kerusakan rel secara lebih detail dan menyeluruh.

Proses konversi video ke gambar terbukti efektif dalam memperkaya variasi dataset. Variasi ini berdampak positif terhadap kemampuan generalisasi model terhadap berbagai kondisi nyata, baik dari sisi pencahayaan, cuaca, maupun sudut pengambilan gambar.

Roboflow berperan besar dalam efisiensi proses anotasi dan pengolahan data. Kemudahan dalam integrasi dengan model YOLO dan fitur augmentasi yang lengkap menjadikan Roboflow sebagai alat bantu yang sangat relevan untuk penelitian ini. Selain itu, proses ekspor dataset yang fleksibel membuat alur pelatihan model lebih terstruktur dan efisien.

Secara keseluruhan, kombinasi metode segmentasi berbasis YOLOv8-seg, dukungan data dari konversi video, serta platform Roboflow mampu menghasilkan sistem deteksi rel kereta yang cepat, akurat, dan efisien. Meski demikian, tantangan seperti ketidakterbacaan citra karena bayangan ekstrem atau objek penghalang tetap perlu diperhatikan untuk pengembangan model ke depannya.

# Assignment

* **Ubah parameter Canny Edge Detection dan bandingkan hasilnya.**

Parameter Threshold 1: (50, 150)

* Deteksi tepi lebih sensitif terhadap perubahan warna dan tekstur.
* Lebih banyak noise terdeteksi, termasuk bagian yang tidak relevan seperti bayangan atau kerikil di sekitar rel.
* Rel tetap terlihat, tetapi ada banyak gangguan yang bisa menghambat segmentasi.

Parameter Threshold 2: (200, 250)

* Hanya tepi yang sangat jelas yang terdeteksi.
* Noise berkurang secara signifikan, tetapi beberapa bagian rel mungkin hilang dari deteksi.
* 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:

* Menggunakan video sebagai sumber dataset dan mengekstrak 1150 frame.
* Frame diambil secara berkala untuk mencakup variasi sudut dan pencahayaan.

Anotasi Dataset:

* Menggunakan Roboflow untuk memberi label jalur rel dalam setiap frame.
* 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.

# Data dan Output Hasil Pengamatan

Data dan hasil yang diperoleh selama percobaan.

|  |  |  |
| --- | --- | --- |
| No | Variabel | Hasil Pengamatan |
| 1 | Hasil Modifikasi model dataset YOLOv8-seg (**rail-dataset.pt**) |  |
| 2 | Hasil Pengujian Ultralytics YOLOv8 Pose Model pada Kamera *Real-Time* |  |
| 3 | Hasil Pengujian Ultralytics YOLOv8 Pose Model pada Video |  |
| 4 | Hasil Pengujian MediaPipe Hands |  |

# Kesimpulan

* Canny Edge Detection mampu mendeteksi tepi jalur rel dengan cepat, tetapi rentan terhadap noise dan tidak dapat membedakan objek berdasarkan konteks.
* Instance Segmentation dengan YOLOv8-seg lebih akurat dalam mendeteksi jalur rel karena mampu mengenali bentuk keseluruhan rel dan membedakannya dari objek lain di sekitar.
* Pemilihan parameter yang optimal bergantung pada kondisi pencahayaan dan kualitas gambar yang digunakan.

# Saran

* Menambahkan lebih banyak variasi kondisi lingkungan (misalnya hujan, kabut, perubahan sudut kamera) untuk meningkatkan generalisasi model (dataset).
* Mencoba **Adaptive Canny Thresholding** agar threshold dapat menyesuaikan dengan kontras gambar secara otomatis.

# Daftar Pustaka

Canny, J. (1986). A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6), 679–698. https://doi.org/10.1109/TPAMI.1986.4767851

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2961–2969). https://doi.org/10.1109/ICCV.2017.322

Jocher, G., Chaurasia, A., Qiu, J., & Stoken, A. (2023). YOLO by Ultralytics. GitHub Repository. <https://github.com/ultralytics/ultralytics>

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4), 834–848. https://doi.org/10.1109/TPAMI.2017.2699184

Li, Y., Song, J., Liu, M., Li, P., & Wang, H. (2022). Intelligent detection of railway track defects using deep learning: A review. IEEE Access, 10, 15979–15991. https://doi.org/10.1109/ACCESS.2022.3149122

Bradski, G. (2000). The OpenCV library. Dr. Dobb's Journal of Software Tools. <https://opencv.org/>

FFmpeg Developers. (2023). FFmpeg documentation. FFmpeg. <https://ffmpeg.org/documentation.html>

Roboflow Inc. (2023). Roboflow: Annotate, preprocess, and train computer vision models. <https://roboflow.com/>