

1. Analisis Simulasi Information Extraction (Python dan OpenCV)

1.1 Ekstraksi Garis dengan Hough Transform

- **Penjelasan Kode:**

- Gambar dimuat dalam format grayscale untuk memudahkan deteksi fitur.
- Deteksi tepi dilakukan menggunakan algoritma Canny dengan parameter batas ambang rendah dan tinggi.
- Transformasi Hough digunakan untuk mendeteksi garis dalam gambar berdasarkan informasi tepi.
- Garis yang terdeteksi digambar di atas gambar asli menggunakan koordinat dari hasil Transformasi Hough.

- **Analisis Hasil:**

- Teknik ini efektif untuk mendeteksi garis lurus yang signifikan dalam gambar.
- Keberhasilan tergantung pada kualitas gambar dan parameter ambang pada algoritma Canny.

1.2 Template Matching untuk Deteksi Objek

- **Penjelasan Kode:**

- Gambar template dan target dimuat dalam mode grayscale.
- Fungsi `cv2.matchTemplate` digunakan untuk mencocokkan template dengan gambar target.
- Lokasi dengan kecocokan tertinggi ditandai dengan kotak.

- **Analisis Hasil:**

- Teknik ini sangat berguna untuk mendeteksi pola tertentu dalam gambar.
- Kelemahannya adalah teknik ini tidak toleran terhadap rotasi atau perubahan skala objek.

1.3 Pembuatan Pyramid Gambar

- **Penjelasan Kode:**

- Gambar diperkecil secara bertahap menggunakan fungsi `cv2.pyrDown` untuk membuat level piramida.
- Setiap level menampilkan versi gambar dengan resolusi lebih rendah.

- **Analisis Hasil:**

- Teknik ini membantu dalam analisis multi-resolusi, seperti deteksi fitur pada skala yang berbeda.
- Resolusi yang lebih rendah dapat kehilangan detail penting dalam gambar asli.

1.4 Deteksi Lingkaran Menggunakan Hough Transform

- **Penjelasan Kode:**

- Gambar diberi efek blur menggunakan GaussianBlur untuk mengurangi noise.
- Transformasi Hough digunakan untuk mendeteksi lingkaran berdasarkan gradien tepi.
- Lingkaran yang terdeteksi digambar di atas gambar asli.

- **Analisis Hasil:**

- Teknik ini efektif untuk mendeteksi lingkaran yang jelas dan bebas noise.
- Noise yang tinggi atau deformasi bentuk dapat mengurangi akurasi deteksi.

1.5 Ekstraksi Warna Dominan

- **Penjelasan Kode:**

- Gambar dikonversi ke format RGB dan diubah menjadi array 2D untuk clustering.
- Algoritma KMeans digunakan untuk mengelompokkan piksel ke dalam beberapa cluster warna.
- Warna dominan ditampilkan berdasarkan centroid cluster.

- **Analisis Hasil:**

- Teknik ini cocok untuk analisis warna utama dalam gambar.
- Akurasi tergantung pada jumlah cluster yang dipilih dan distribusi warna dalam gambar.

1.6 Deteksi Kontur pada Gambar

- **Penjelasan Kode:**

- Deteksi tepi dilakukan dengan algoritma Canny untuk menemukan batas objek.
- Fungsi `cv2.findContours` digunakan untuk mendeteksi kontur berdasarkan tepi yang terdeteksi.
- Kontur digambar di atas gambar asli.

- **Analisis Hasil:**

- Teknik ini sangat berguna untuk mendeteksi bentuk dan batas objek.
- Akurasi bergantung pada kualitas deteksi tepi dan parameter pada algoritma Canny.

2. Analisis Simulasi Webots

2.1 Lidar Data Extraction

- **Penjelasan Kode:**
 - Sensor Lidar diaktifkan dan dikonfigurasi untuk mendapatkan data rentang jarak dalam bentuk array.
 - Fungsi `lidar.getRangeImage` digunakan untuk membaca data jarak dari lingkungan.
 - Data Lidar dianalisis untuk menentukan keberadaan objek di berbagai arah.
- **Analisis Hasil:**
 - Data Lidar memberikan informasi jarak yang akurat dari lingkungan sekitar robot.
 - Informasi ini penting untuk navigasi dan penghindaran rintangan.

2.2 Obstacle Detection

- **Penjelasan Kode:**
 - Sensor jarak ultrasonik digunakan untuk mendeteksi objek di depan robot.
 - Data jarak dari sensor digunakan untuk menghitung kecepatan motor kiri dan kanan berdasarkan koefisien penghindaran tabrakan.
 - Fungsi `compute_speeds` mengatur kecepatan motor untuk menghindari rintangan.
- **Analisis Hasil:**
 - Teknik ini berhasil mendeteksi dan menghindari rintangan secara real-time.
 - Efektivitas bergantung pada penempatan sensor dan parameter koefisien yang digunakan.