

Tugas 1: Ekstraksi Informasi Gambar dengan OpenCV

Tugas ini berfokus pada penerapan teknik ekstraksi informasi dari gambar menggunakan OpenCV di Google Colab, yang terdiri dari beberapa tahapan berikut:

1. Ekstraksi Garis dengan Hough Transform:

- **Hough Transform** adalah teknik yang digunakan untuk mendeteksi garis dalam gambar. Pada dasarnya, garis yang terdeteksi dalam ruang gambar diubah ke ruang parameter (ruang Hough). Dengan memetakan titik-titik yang membentuk garis, kita dapat mendeteksi garis lurus meskipun gambar mengandung noise.
- Proses ini sering diawali dengan deteksi tepi menggunakan algoritma Canny, lalu menggunakan `cv2.HoughLines()` untuk mendeteksi garis-garis yang ada.

2. Template Matching untuk Deteksi Objek:

- **Template Matching** adalah metode untuk menemukan lokasi objek tertentu dalam gambar. Dengan membandingkan gambar dengan template yang sudah diketahui, kita dapat menemukan posisi yang mirip dengan template tersebut.
- Proses ini melibatkan pencocokan gambar dengan template menggunakan cross-correlation untuk menemukan area yang paling mirip dengan template yang diberikan.

3. Pembuatan Pyramid Gambar:

- Pyramid gambar digunakan untuk mengubah resolusi gambar ke berbagai tingkat, yang memungkinkan deteksi objek pada berbagai skala. Teknik ini berguna dalam banyak aplikasi pengolahan gambar, seperti deteksi objek atau fitur yang bervariasi ukurannya.
- Pada OpenCV, ini dapat dilakukan dengan fungsi `cv2.pyrDown()` untuk mengecilkan gambar dan `cv2.pyrUp()` untuk memperbesar gambar.

4. Deteksi Lingkaran Menggunakan Hough Transform:

- **Hough Transform untuk Lingkaran** digunakan untuk mendeteksi lingkaran dalam gambar, mirip dengan deteksi garis. Pada OpenCV, ini bisa dilakukan dengan fungsi `cv2.HoughCircles()`, yang memanfaatkan deteksi tepi dan ruang Hough untuk menemukan lingkaran berdasarkan parameter radius dan posisi.

5. Ekstraksi Warna Dominan pada Gambar:

- **Ekstraksi warna dominan** berfokus pada mengambil warna utama dalam gambar, biasanya menggunakan ruang warna HSV (Hue, Saturation, Value). Teknik ini digunakan untuk mendeteksi warna yang paling dominan di dalam gambar, yang sering digunakan dalam pengolahan citra dan analisis visual.

- Histogram warna digunakan untuk menghitung distribusi warna, dan nilai dengan frekuensi tertinggi akan dianggap sebagai warna dominan.

6. **Deteksi Kontur pada Gambar:**

- **Deteksi Kontur** memungkinkan untuk mendeteksi bentuk objek dalam gambar dengan mencari batas-batas objek. Kontur yang ditemukan dapat digunakan untuk analisis bentuk atau pemisahan objek.
- Pada OpenCV, ini dilakukan dengan fungsi `cv2.findContours()` yang menghasilkan kontur sebagai garis batas objek dalam gambar.

Tugas 2: LIDAR Data Extraction and Obstacle Detection

Tugas kedua berfokus pada penggunaan data LIDAR dan sensor jarak untuk mendeteksi dan menghindari rintangan dalam simulasi robot. Berikut adalah langkah-langkah yang dilakukan:

1. **Inisialisasi Sensor LIDAR dan Ultrasonic:**

- Sensor **LIDAR** digunakan untuk mengumpulkan data jarak dari objek di sekitar robot dengan memanfaatkan pemindaian laser. LIDAR menghasilkan data dalam bentuk gambar jarak yang dapat memberikan informasi tentang posisi dan jarak rintangan.
- Sensor **ultrasonic** memberikan data jarak yang lebih langsung ke objek di dekatnya, yang digunakan untuk mendeteksi rintangan lebih tepat.

2. **Pengaturan Motor:**

- Motor diatur dalam mode kecepatan untuk menggerakkan robot dengan kecepatan tertentu, yang dapat disesuaikan berdasarkan informasi dari sensor.
- Motor kiri dan kanan diatur dengan posisi tak terbatas, yang memungkinkan mereka bekerja dalam mode kecepatan konstan, dan kecepatannya dihitung berdasarkan informasi sensor yang diperoleh.

3. **Penghindaran Tabrakan Berdasarkan Data Sensor:**

- **LIDAR** digunakan untuk mengidentifikasi rintangan pada jarak yang lebih jauh, sedangkan **sensor ultrasonic** memberikan informasi jarak lebih langsung ke objek di sekitar robot.
- Berdasarkan data yang diterima dari sensor, robot menghitung kecepatan roda kiri dan kanan untuk menghindari tabrakan. Misalnya, jika sensor mendeteksi rintangan di sebelah kiri, motor kiri akan diperlambat untuk menghindari tabrakan.

4. Pemrograman Kecepatan dengan Koefisien Empiris:

- Koefisien empiris digunakan dalam rumus untuk mengubah nilai sensor menjadi perubahan kecepatan roda. Hal ini memungkinkan robot untuk bergerak dengan cara yang lebih responsif terhadap rintangan yang terdeteksi.
- Misalnya, jika ada objek di sebelah kiri robot, kecepatan roda kiri akan dikurangi, sedangkan kecepatan roda kanan akan tetap lebih tinggi untuk memungkinkan robot berbelok atau menghindari.

5. Proses Ekstraksi Data dan Kontrol Penghindaran:

- **Fungsi Ekstraksi Data LIDAR:** Fungsi `extract_lidar_data()` mengumpulkan gambar jarak dari LIDAR dan menampilkan data dari berbagai sudut. Ini memungkinkan sistem untuk mengenali potensi rintangan dan menghitung jalur yang harus diambil.
- **Pembacaan Sensor Jarak:** Fungsi `read_distance_sensors()` mengumpulkan data jarak dari sensor ultrasonic dan menampilkan hasilnya untuk digunakan dalam pengambilan keputusan.

6. Kontrol Kecepatan Berdasarkan Sensor:

- Fungsi `compute_speeds()` menggunakan pembacaan sensor ultrasonic untuk menghitung kecepatan roda berdasarkan data yang ada. Dengan demikian, robot dapat menyesuaikan kecepatannya untuk menghindari tabrakan dan bergerak dengan lancar.

Kesimpulan:

- **Tugas 1:** Menggunakan OpenCV di Google Colab untuk ekstraksi informasi gambar melalui beberapa teknik, seperti deteksi objek dengan template matching, deteksi garis dan lingkaran dengan Hough Transform, ekstraksi warna dominan, dan deteksi kontur untuk mempelajari lebih lanjut mengenai pengolahan citra.
- **Tugas 2:** Simulasi robot menggunakan LIDAR dan sensor ultrasonic untuk mendeteksi dan menghindari rintangan dengan menggunakan kecepatan roda yang dihitung berdasarkan data sensor, memberikan kontrol penghindaran tabrakan berbasis jarak yang sederhana dan efektif.