Hasil Analisis

1. JetBot Basic Motion (Simulasi Pergerakan Dasar)

Deskripsi:

- Pada simulasi ini, JetBot diberi instruksi untuk melakukan pergerakan dasar seperti maju, mundur, berbelok kiri, dan berbelok kanan.
- Pergerakan dikendalikan dengan mengatur kecepatan motor kiri dan kanan menggunakan fungsi left(), right(), forward(), backward(), dan stop().
- Simulasi ini berjalan dalam loop di mana robot akan melakukan pergerakan maju, mundur, kiri, dan kanan secara berulang dengan interval waktu tertentu.

Analisis:

- Tujuan: Memberikan pemahaman dasar tentang kontrol motor dan pergerakan JetBot di dunia simulasi Webots.
- Implementasi: Penggunaan API Webots (wb_motor_set_velocity) untuk mengatur kecepatan motor kiri dan kanan memungkinkan kontrol pergerakan robot dengan respons yang cukup cepat.
- Kelebihan: Kode ini sederhana dan efektif untuk memulai dengan pergerakan dasar JetBot.
- Kekurangan: Pergerakan yang dilakukan masih bersifat manual tanpa adanya kecerdasan buatan (AI) yang mendukung. Robot tidak dapat menghindari halangan atau memutuskan arah berdasarkan kondisi lingkungan.

2. JetBot Collect Data (Pengumpulan Data untuk Pelatihan Model)

Deskripsi:

- Dalam simulasi ini, JetBot dilatih untuk mengumpulkan gambar dari kamera dalam dua kategori: "free" (jalur bebas) dan "blocked" (jalur terhalang).
- Setiap kali pengguna menekan tombol "F" atau "B", gambar disimpan ke dalam folder yang sesuai berdasarkan kategori yang dipilih.
- Setelah mengumpulkan data, model pelatihan dapat dijalankan menggunakan dataset gambar yang telah dikumpulkan untuk melatih model AI dalam mengklasifikasikan jalur bebas atau terhalang.

Analisis:

- Tujuan: Menyediakan dataset gambar yang dibutuhkan untuk melatih model penghindaran tabrakan JetBot, yang kemudian digunakan untuk klasifikasi jalur berdasarkan input kamera.
- Implementasi: Fungsi pengumpulan data dan penyimpanan gambar sudah berjalan dengan baik. Menggunakan PIL.Image untuk menangani gambar yang diambil dari kamera robot, yang kemudian disimpan dalam folder terpisah.

- Kelebihan: Data yang terkumpul dapat digunakan untuk pelatihan model AI dan mempercepat proses pembuatan sistem penghindaran tabrakan berbasis visi komputer.
- Kekurangan: Proses pengumpulan data masih bergantung pada input manual dari pengguna untuk menentukan apakah jalur yang terdeteksi adalah "free" atau "blocked". Ini mungkin tidak ideal untuk situasi di mana pengumpulan data otomatis diperlukan.

3. JetBot Collision Avoidance (Penghindaran Tabrakan dengan Model AI)

Deskripsi:

- Setelah model pelatihan selesai, simulasi ini menggunakan model ResNet18 yang telah dilatih untuk menganalisis gambar dari kamera JetBot dan memutuskan apakah robot dapat bergerak maju atau harus berbelok.
- Jika model mendeteksi jalur bebas (probabilitas < 0.5 untuk "blocked"), robot akan bergerak maju. Jika tidak, robot akan berbelok ke kiri untuk menghindari tabrakan.

Analisis:

- Tujuan: Mengimplementasikan sistem penghindaran tabrakan yang didasarkan pada model AI yang dilatih menggunakan dataset yang dikumpulkan.
- Implementasi: Proses implementasi melibatkan pemrosesan gambar dari kamera JetBot, normalisasi gambar menggunakan torchvision.transforms.Normalize, dan kemudian mengklasifikasikan gambar menggunakan model neural network (ResNet18). Model ini menggunakan softmax untuk menghasilkan probabilitas dan membuat keputusan berdasarkan output tersebut.
- Kelebihan: Integrasi model AI dengan JetBot memberikan kontrol yang lebih cerdas dan otomatis untuk navigasi robot, memungkinkan JetBot untuk menghindari halangan dengan memanfaatkan visi komputer.
- Kekurangan: Model ResNet18 cukup sederhana dan mungkin memerlukan lebih banyak pelatihan untuk dapat berfungsi dengan baik dalam kondisi nyata atau lebih kompleks. Selain itu, ketergantungan pada model yang dilatih dengan dataset terbatas dapat menyebabkan kinerja yang buruk jika ada variasi besar dalam lingkungan.