Nama : Ketut Satria Wibisana

NIM : 1103213148

Kelas : TK-45-G09

Analisis Simulasi Image Processing dan Feature Detection dengan Python dan OpenCV

1. **Filter Moving Average**

   Filter moving average adalah metode smoothing yang sederhana, yang menghitung rata-rata nilai intensitas piksel dalam sebuah kernel berbentuk persegi panjang. Teknik ini digunakan untuk mengurangi noise yang seragam, sehingga menghasilkan gambar yang lebih halus meskipun dengan detail yang lebih sedikit. Dalam simulasi, filter ini diterapkan menggunakan fungsi `cv2.filter2D()`. Dengan ukuran kernel k x k, filter ini dapat membuat gambar tampak buram namun efektif dalam menghilangkan noise pada tingkat rendah. Namun, kelemahan utamanya adalah pengurangan ketajaman pada tepi objek, yang dapat mempengaruhi analisis lanjutan seperti deteksi fitur.

2. **Deteksi Fitur dengan SIFT (Scale-Invariant Feature Transform)**

   SIFT digunakan untuk mendeteksi dan mendeskripsikan fitur-fitur dalam gambar dengan ketahanan terhadap perubahan skala, rotasi, dan pencahayaan. Dalam simulasi, OpenCV menyediakan fungsi `cv2.SIFT_create() yang memungkinkan deteksi keypoints serta pembuatan deskriptor fitur. Saat diterapkan pada gambar yang kompleks seperti pemandangan atau objek bertekstur, SIFT mampu mengenali fitur-fitur penting seperti sudut, ujung tekstur, dan pola yang khas. Hasil simulasi menunjukkan bahwa SIFT sangat handal dalam melakukan tugas seperti mencocokkan objek pada gambar yang mengalami rotasi atau skala perubahan.

3. **Representasi Histogram Gambar**

   Histogram digunakan untuk menganalisis distribusi piksel dalam sebuah gambar. Dengan menggunakan fungsi `cv2.calcHist()`, kita dapat menghasilkan histogram baik untuk saluran warna tertentu maupun untuk gambar grayscale. Representasi histogram ini memberikan informasi mengenai karakteristik visual gambar, seperti distribusi kecerahan atau dominasi warna tertentu. Dalam aplikasi praktis, histogram dimanfaatkan untuk peningkatan kontras, seperti proses equalization, yang bertujuan membuat gambar lebih jelas. Simulasi ini menunjukkan betapa pentingnya histogram dalam tahap pemrosesan awal maupun dalam analisis yang lebih mendalam.

4. **Gaussian Smoothing**

   Gaussian smoothing menggunakan kernel berbobot Gaussian untuk mengurangi noise sekaligus mempertahankan detail tepi dengan lebih baik dibandingkan filter moving

average. Penerapan metode ini melalui fungsi `cv2.GaussianBlur()` dalam simulasi menunjukkan efek penyamaran yang lebih alami, berkat bobot kernel yang menurun seiring dengan peningkatan jarak dari pusat. Simulasi tersebut mengilustrasikan manfaat Gaussian smoothing dalam tahap preprocessing, khususnya dalam mempersiapkan gambar sebelum melakukan deteksi fitur atau segmentasi objek.

5. **Deteksi Tepi dengan Sobel Filter**

   Filter Sobel merupakan metode deteksi tepi yang berbasis pada gradien intensitas. Dengan memanfaatkan kernel khusus untuk menghitung variasi intensitas pada arah xxx dan yyy, filter Sobel mampu mengenali tepi horizontal, vertikal, maupun kombinasi keduanya. Dalam simulasi, fungsi `cv2.Sobel()` menghasilkan gambar yang menonjolkan tepi-tepi tersebut. Filter ini sangat penting untuk proses segmentasi atau analisis bentuk, namun memiliki kelemahan berupa kerentanan terhadap noise jika gambar belum dihaluskan terlebih dahulu.

6. **Representasi Fitur dengan HOG (Histogram of Oriented Gradients)**

   HOG (Histogram of Oriented Gradients) adalah teknik deskripsi fitur yang memanfaatkan pola penyebaran gradien orientasi dalam suatu citra. Metode ini terbukti sangat efektif untuk mendeteksi objek seperti manusia atau kendaraan. Dengan OpenCV, perhitungan HOG dilakukan dengan menghitung gradien orientasi pada setiap sel piksel, lalu hasilnya diringkas dalam bentuk histogram. Dari hasil simulasi, dapat dilihat bahwa HOG memiliki sensitivitas tinggi terhadap pola-pola lokal pada gambar, sehingga sangat ideal untuk pengenalan objek yang berbasis pada tekstur maupun bentuk.

Analisis Simulasi Robot dengan Webots dan OpenCV

1. **Visual Tracking dengan OpenCV**

   Visual tracking merupakan metode untuk mengikuti pergerakan objek secara real-time. Dalam simulasi ini, bola berwarna merah dilacak dengan cara mengonversi gambar ke ruang warna HSV menggunakan fungsi cv2.cvtColor(), kemudian menerapkan thresholding untuk menonjolkan warna yang diinginkan. Masker biner yang dihasilkan menunjukkan lokasi objek dalam setiap frame. Posisi centroid dihitung menggunakan cv2.moments(), dan sebuah pengendali P sederhana digunakan untuk mengatur gerakan robot berdasarkan perbedaan posisi bola dari titik tengah frame. Pendekatan ini memiliki kelebihan dalam hal kesederhanaannya, namun kepekaannya terhadap perubahan pencahayaan atau variasi warna latar belakang bisa menjadi suatu kendala.

2. **Document Scanner Simulation**

   Simulasi ini melibatkan beberapa tahap utama dalam pengolahan citra:

   - Pra-pemrosesan: Menggunakan thresholding adaptif dengan cv2.adaptiveThreshold() untuk menghasilkan gambar biner yang menonjolkan dokumen.

- Deteksi Kontur: Memanfaatkan cv2.findContours() untuk menemukan kontur dokumen, yang kemudian diidentifikasi sebagai persegi panjang terbesar dalam frame.
- Transformasi Perspektif: Mengaplikasikan cv2.getPerspectiveTransform() untuk melakukan transformasi warp sehingga dokumen ditampilkan dalam perspektif atas-bawah.

Simulasi ini menyerupai aplikasi pemindai pada ponsel dan sangat efektif untuk mendigitalkan dokumen secara efisien. Namun, terdapat kendala pada proyek ini dimana tekstur kotak atau kardus berukuran besar tidak ter-load dengan baik, sehingga Document Scanner tidak berfungsi untuk kotak berukuran besar, meskipun berfungsi dengan baik untuk kotak berukuran kecil.

3. Fruit Detection Robot

Simulasi deteksi buah mengombinasikan teknologi computer vision dengan pengendalian robotik secara efektif. Berikut adalah langkah-langkah utama dalam simulasi ini:

- Identifikasi Buah: Warna buah dianalisis dalam ruang warna HSV. Proses thresholding diterapkan untuk masing-masing warna buah tertentu, seperti merah untuk apel atau kuning untuk pisang. Kontur buah kemudian diekstraksi menggunakan cv2.findContours().
- Penentuan Lokasi: Koordinat centroid dari setiap buah dihitung, yang menyediakan informasi posisi untuk pengendalian robot.
- Pengendalian Lengan Robot: Koordinat yang diperoleh digunakan untuk mengarahkan lengan robot dengan presisi menuju lokasi buah dan melakukan pemindahan.

Dalam simulasi ini, OpenCV berperan sebagai alat utama untuk pengenalan dan identifikasi objek, sementara pengendali robot bertanggung jawab atas manipulasi fisik objek tersebut.