

Laporan Praktikum Kontrol Cerdas

Minggu ke-1

Nama : Kresensia Meita Indar Mayaningsih

NIM : 224308087

Kelas : TKA 7D

Akun Github (Tautan) : <https://github.com/kzmeita>

Student Lab Assistant :

1. Judul Percobaan :

Deteksi dan Klasifikasi Warna Objek Menggunakan Convolutional Neural Network (CNN) dan OpenCV

2. Tujuan Percobaan

Tujuan dari praktikum minggu ke-1 :

1. Mahasiswa dapat memahami konsep dasar sistem kontrol cerdas
2. Mahasiswa dapat memahami konsep dasar Convolutional Neural Network (CNN)
3. Mahasiswa dapat memahami cara mendeteksi warna objek menggunakan OpenCV
4. Mahasiswa dapat mengimplementasikan Convolutional Neural Network (CNN) untuk klasifikasi warna objek
5. Mahasiswa dapat menghasilkan sistem yang mampu mengidentifikasi warna objek secara akurat dari gambar atau video.
6. Mahasiswa dapat memahami penerapan komputer vision dalam pengolahan citra digital

3. Landasan Teori

Kecerdasan Buatan (AI) adalah cabang ilmu komputer yang berfokus pada pembuatan sistem atau program yang mampu meniru kemampuan berpikir, belajar, dan mengambil keputusan seperti manusia. Menurut (MF AK, 2021) bahwa kecerdasan buatan menandai perubahan zaman ke zaman digital, dimana peran teknologi sangat besar dalam kehidupan. Kemajuan pesat dalam bidang kecerdasan buatan, khususnya dalam pengenalan objek berbasis gambar, telah membuka peluang baru untuk mengembangkan sistem otomatis untuk mengidentifikasi jenis warna, pola, atau garis yang berbeda (Nengsih, W. & Yulina, S., 2024). Deep Learning adalah bagian dari kecerdasan buatan dan machine learning yang merupakan pengembangan dari neural network multiple layer

untuk memberikan ketepatan tugas seperti deteksi objek, pengenalan suara, terjemahan bahasa dan lainnya (Raup, A. et al., 2022). Dalam pengolahan citra, Deep Learning dapat digunakan untuk mendeteksi, mengenali, dan mengklasifikasikan objek secara otomatis.

CNN merupakan operasi konvolusi yang menggabungkan beberapa lapisan pemrosesan, menggunakan beberapa elemen yang beroperasi secara paralel dan terinspirasi oleh sistem saraf biologi (Prabowo, dkk., 2024). Convolutional Neural Networks (CNN) adalah metode dalam deep learning yang sering digunakan untuk mengolah data berbentuk grid seperti gambar. CNN telah menunjukkan hasil yang menjanjikan dalam berbagai aplikasi pengolahan citra (Pradana, dkk., 2024). CNN mampu mendeteksi pola dan fitur secara otomatis dengan presisi tinggi melalui serangkaian lapisan konvolusi, pooling, dan fully connected (Taye, 2023; Yamashita, 2018). Pada CNN terdapat jenis lapisan diantaranya lapisan konvolusi (Convolution Layer) yang bertugas mengekstraksi fitur dari gambar, seperti tepi, warna, dan tekstur; lapisan pooling (Pooling Layer) digunakan untuk mengurangi dimensi data sekaligus mempertahankan informasi penting, sehingga mempercepat proses komputasi; dan lapisan Fully Connected yang menghasilkan output klasifikasi akhir berdasarkan fitur yang telah diekstraksi. Dengan CNN, analisis gambar menjadi lebih akurat dan efisien, sehingga sangat sesuai untuk aplikasi deteksi dan klasifikasi warna objek.

Dengan semakin berkembangnya teknologi, implementasi deteksi menggunakan bahasa pemrograman Python melalui pustaka OpenCV menjadi pilihan populer karena kemudahan dalam pengembangan dan fleksibilitasnya (Qisthiano, M. R. & Pratiwi, A. O., 2025). Python sebagai alat untuk pengolahan citra telah menarik perhatian karena keunggulannya dalam ketersediaan berbagai pustaka dan framework yang kuat seperti OpenCV, scikit-image, dan PIL (Python Imaging Library) (Abdusysyaid, F., dkk., 2024.). OpenCV adalah pustaka open-source yang dirancang khusus untuk pengolahan citra dan video. Pustaka ini menyediakan berbagai algoritma untuk deteksi pola visual, seperti fitur-fitur khas wajah manusia, termasuk mata, hidung, dan mulut (Silalahi, D. M. et al., 2024).

Pada praktikum ini, menggunakan bahasa pemrograman python dan pustaka OpenCV sebagai metode untuk mendeteksi warna suatu obyek. Dengan OpenCV, gambar dapat diproses menjadi format yang siap digunakan sebagai input untuk model CNN, misalnya melalui konversi ruang warna dari BGR ke HSV atau ekstraksi masker warna tertentu. Deteksi warna adalah proses identifikasi piksel atau area pada citra yang memiliki nilai warna tertentu. Dalam aplikasi ini, OpenCV digunakan untuk mengubah citra ke ruang warna HSV, yang lebih stabil terhadap perubahan pencahayaan dan membuat masking berdasarkan rentang warna tertentu, sehingga hanya objek dengan warna yang diinginkan

yang dapat diolah. Selanjutnya, CNN digunakan untuk mengklasifikasikan objek berdasarkan fitur warna dan pola yang diekstraksi. Kombinasi OpenCV dan CNN memungkinkan sistem mendeteksi warna secara otomatis sekaligus mengenali objek, sehingga dapat diaplikasikan pada berbagai bidang, seperti robotika, pengawasan industri, dan pendidikan (Prasetyo, 2023; Qisthiano, 2025).

Dalam praktikum ini, model warna BGR (Blue, Green, Red) digunakan sebagai representasi warna default pada OpenCV. Setiap piksel gambar direpresentasikan sebagai kombinasi tiga komponen warna: biru (B), hijau (G), dan merah (R). RGB merupakan model warna dasar yang umum digunakan untuk penyimpanan dan pemrosesan citra digital (Goenawan dkk., 2022). Model ini digunakan sebagai input awal untuk proses masking warna dan klasifikasi dengan CNN, sebelum konversi ke ruang warna lain jika diperlukan (misal HSV untuk kestabilan terhadap pencahayaan). Pada setiap piksel, warna direpresentasikan oleh nilai intensitas dari masing-masing komponen R, G, dan B, dengan rentang nilai dari 0 hingga 255. Setiap titik pada ruang warna RGB merepresentasikan kombinasi unik dari ketiga komponen tersebut. Misalnya titik (0,0,0) menghasilkan warna hitam, sedangkan titik (255,255,255) menghasilkan warna putih. Dengan memahami model RGB, sistem pengolahan citra dapat melakukan deteksi dan klasifikasi warna objek secara akurat, karena setiap warna objek dapat diuraikan menjadi kombinasi nilai R, G, dan B.

4. Analisis dan Diskusi

A. Analisis

Pada praktikum ini, citra objek diproses menggunakan OpenCV dengan model warna BGR, kemudian dianalisis dan diklasifikasikan menggunakan Convolutional Neural Network (CNN). Hasil pengolahan menunjukkan bahwa CNN mampu mengenali dan mengklasifikasikan objek berdasarkan warna dengan akurasi tinggi, terutama pada warna yang memiliki kontras jelas. Beberapa pengamatan penting:

1. Ketepatan Deteksi: Objek dengan warna primer (merah, hijau, biru) terdeteksi lebih konsisten dibandingkan warna campuran atau pastel, karena komponen R, G, B lebih dominan.
2. Pengaruh Pencahayaan: Variasi cahaya memengaruhi intensitas warna pada model BGR, sehingga beberapa objek dengan warna serupa mengalami sedikit pergeseran deteksi. Konversi ke ruang warna HSV dapat menjadi alternatif untuk mengurangi efek pencahayaan.

3. Ukuran dan Posisi Objek: CNN dapat mendeteksi objek dengan ukuran berbeda, namun objek yang terlalu kecil atau sebagian terhalang kadang menghasilkan kesalahan klasifikasi.

Selain itu, analisis terhadap CNN yang digunakan untuk mengekstraksi fitur visual dari citra dan mengklasifikasikan objek berdasarkan kombinasi fitur warna dan pola:

1. Lapisan konvolusi berhasil menyorot bagian objek yang menonjol, seperti tepi dan area dengan intensitas warna tinggi.
2. Lapisan pooling membantu mengurangi ukuran data sambil mempertahankan fitur penting, sehingga mempercepat proses komputasi.
3. Lapisan fully connected menghasilkan prediksi akhir yang digunakan untuk menentukan kategori warna objek.

Hasil praktikum menunjukkan bahwa kombinasi CNN dan OpenCV efektif untuk deteksi warna, terutama bila citra telah melalui tahap masking untuk mengekstrak komponen warna yang diinginkan.

B. Diskusi

Beberapa hal yang perlu diperhatikan berdasarkan hasil praktikum:

1. Akurasi tinggi dicapai ketika objek memiliki warna jenuh dan pencahayaan stabil.
2. Kesalahan klasifikasi terjadi pada warna yang mirip atau pencahayaan rendah, sehingga perlu penyesuaian threshold warna atau augmentasi data.
3. Penggunaan model warna BGR pada OpenCV sangat praktis untuk input CNN, tetapi konversi ke HSV atau LAB dapat meningkatkan kestabilan deteksi terhadap pencahayaan.

Secara keseluruhan, praktikum ini membuktikan bahwa integrasi OpenCV untuk pre-processing citra dengan CNN untuk klasifikasi merupakan metode yang efisien dan akurat untuk deteksi dan klasifikasi warna objek.

5. Assignment

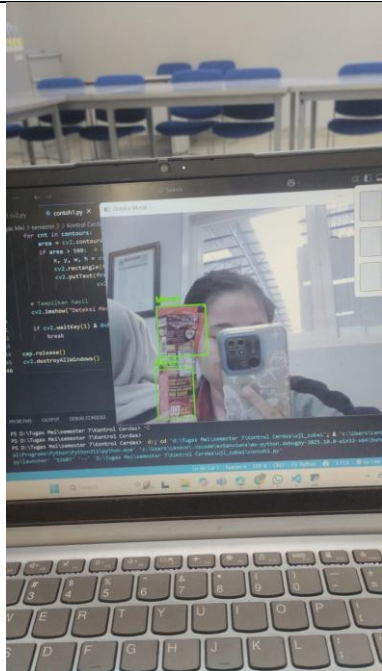
Pada praktikum ini, program deteksi warna dikembangkan dari versi sebelumnya yang hanya mampu mendeteksi warna merah. Assignment ini menambahkan kemampuan untuk mendeteksi tiga warna sekaligus, yaitu merah, biru, dan hijau, serta menambahkan bounding box dengan label warna di atas objek yang terdeteksi.

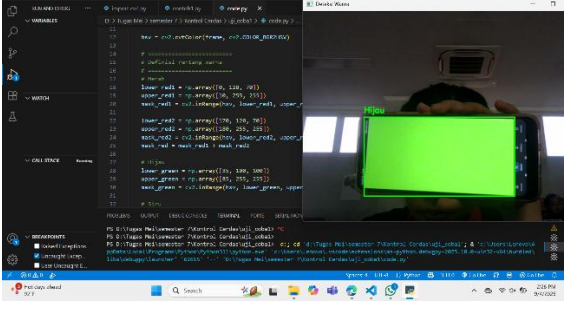
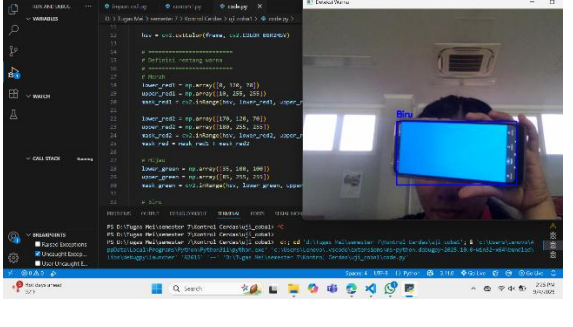
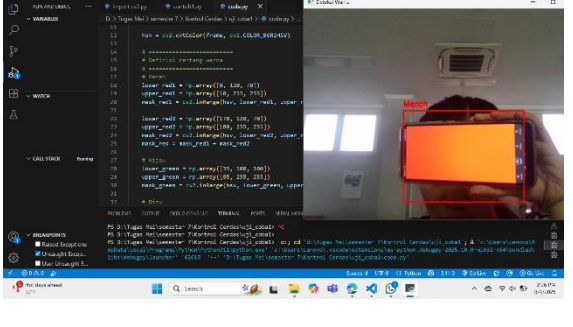
Beberapa perubahan dan fitur penting pada assignment ini:

- a. Inisialisasi Kamera: Kamera dibuka menggunakan OpenCV (`cv2.VideoCapture(0)`) untuk menangkap citra secara real-time.
- b. Penggunaan HSV: Citra dari kamera dikonversi dari model warna BGR ke HSV menggunakan `cv2.cvtColor` untuk mempermudah proses masking warna.
- c. Penggabungan Masking: Masking dilakukan untuk ketiga warna merah (Dua rentang hue (0–10 dan 170–180) digabung untuk menangkap seluruh spektrum merah), biru (Rentang hue 90–130), dan hijau (Rentang hue 35–85) digabungkan menjadi satu frame.
- d. Masking Warna: `cv2.inRange()` digunakan untuk membuat mask untuk setiap warna. Mask ini menandai piksel yang termasuk warna target.
- e. Tampilan Hasil: Hasil deteksi ditampilkan dalam satu jendela real-time (`cv2.imshow("Deteksi Warna", frame)`), sehingga lebih sederhana dan informatif.
- f. Deteksi dan Bounding Box: Fungsi `detect_color()` digunakan untuk:
 - Menemukan kontur objek dengan `cv2.findContours`.
 - Memberikan bounding box pada objek (`cv2.rectangle`).
 - Menambahkan label warna di atas objek (`cv2.putText`).
 - Hanya objek dengan area > 500 piksel yang diberi kotak untuk menghindari noise.

6. Data dan Output Hasil Pengamatan

Data yang diperoleh pada praktikum minggu ke-1 disajikan dalam tabel dibawah ini:

No	Variabel	Hasil Pengamatan
1.	Code sebelum modifikasi hanya dapat mendeteksi 1 warna saja, disini menggunakan warna merah sebagai percobaan pertama. Pada percobaan pertama ini, program dapat mendeteksi warna merah namun program belum mendeteksi secara tepat karena terdapat beberapa warna merah yang tidak terdeteksi. Perlu di lakukan pelatihan secara continue sehingga program dapat mendeteksi warna secara akurat dan real-time.	

2.	Code setelah dimodifikasi dapat membaca 3 warna sekaligus yaitu hijau, biru, dan merah. Pada data output nomor 2 ini, mencoba mendeteksi warna hijau dan terlihat program berhasil mendeteksi warna hijau.	
3.	Code setelah dimodifikasi ini mencoba mendeteksi warna biru dan program berhasil mendeteksi dengan benar sesuai dengan sintak.	
4.	Program mencoba mendeteksi warna merah dan berhasil mendeteksi warna dengan benar dan akurat sesuai dengan sintak yang diperintahkan.	

7. Kesimpulan

Setelah melakukan praktikum mengenai sistem pendeteksi warna menggunakan CNN dan OpenCV dapat ditarik kesimpulan antara lain:

1. Praktikum ini berhasil membuktikan bahwa OpenCV dapat digunakan untuk mendeteksi warna objek secara real-time dengan menggunakan model warna HSV.
2. Dengan pengembangan assignment, yang tadinya hanya dapat mendeteksi satu warna saja program kini mampu mendeteksi tiga warna sekaligus (merah, hijau, biru) dan menampilkan bounding box serta label warna, sehingga visualisasi menjadi lebih jelas dan informatif.
3. Fungsi masking warna dan deteksi kontur memungkinkan sistem membedakan objek berdasarkan warna meskipun berada di lingkungan dengan pencahayaan bervariasi, meski tetap terdapat keterbatasan pada warna yang mirip atau objek yang terlalu kecil.
4. Kombinasi OpenCV dan Python terbukti efektif untuk pengolahan citra sederhana, pengenalan warna, dan visualisasi objek, sehingga dapat dijadikan dasar untuk aplikasi lebih kompleks, seperti robotika, sistem pengawasan, atau pendidikan.

5. Tantangan yang ditemukan antara lain Deteksi objek bisa terganggu oleh latar belakang dengan warna mirip target dan Objek kecil atau sebagian tersembunyi mungkin tidak terdeteksi.

8. Saran

1. Peningkatan Akurasi Deteksi dengan melakukan kalibrasi rentang HSV sesuai kondisi pencahayaan di lokasi praktikum untuk meningkatkan akurasi deteksi.
2. Pengolahan Objek Kecil yang tersembunyi sulit terdeteksi; disarankan menyesuaikan nilai threshold area kontur atau menambahkan metode filtering agar objek kecil tetap terdeteksi.
3. Optimasi Kode dengan Fungsi deteksi warna bisa dibuat lebih modular untuk menambahkan warna lain atau fitur tambahan tanpa mengubah struktur utama program dan menambahkan informasi tambahan pada frame, seperti jumlah objek yang terdeteksi atau FPS (frame per second), untuk analisis lebih lengkap.
4. Penggunaan Kamera dalam kondisi baik dan tidak digunakan aplikasi lain agar tidak terjadi error “Camera index out of range” dan bisa menambahkan kode pengecekan otomatis untuk memastikan kamera tersedia sebelum program berjalan.

9. Daftar Pustaka

- Silalahi, D., Putra, D., Sari, D., Pramudya, F., & Supiyandi. (2024). Aplikasi Python untuk Deteksi Wajah pada Frame Menggunakan Algoritma Haar. *Jurnal Komputer dan Aplikasi*, 12, 298-310.
- Abdusysyaid, F., F, I., & S, C. (2024). Penggunaan Python Sebagai Pengolahan Citra Untuk Mencari Ukuran Partikel Dari Suatu Gambar Mikroskop Elektron. *e-Proceeding of Engineering*, 11, 6615.
- MF AK. (2021). Pembelajaran Digital. Bandung : Widina Bhakti Persada.
- Nengsih , W., & Yulina , S. (2024). Implementasi Model CNN Untuk Identifikasi Jenis Bunga Berdasarkan Spektrum Warna. *Jurnal Komputer Terapan*, 10, 57 - 66.
- Prabowo, F. W., Homaidi, A., & Lutfi, A. (2024). Deteksi Warna Kulit Menggunakan Metode Deep Learning Dengan CNN (Convolutional Neural Network) Untuk Menentukan Kecocokan Warna Kulit dan Warna Busana. *Jurnal Teknik Elektro dan Informatika*, 19, 186-190 .
- Pradana, I. A., Rahajoe, A., & Sihananto, A. (2024). Pengembangan Aplikasi Pendeteksi Keretakan Jalan Berbasis Android Dengan Implementasi Algoritma Hybrid CNN-ISTM. *Jurnal Informatika dan Sistem Informasi* , 5, 2722-130.
- Prasetyo, D. D. (2023). *OpenCV untuk pengolahan citra dan visi komputer*. Community Algotstudio.

Qisthiano, M., & Pratiwi, A. (2025). Deteksi Tepi Pada Citra Objek Benda Menggunakan Algoritma Sobel Dan Prewitt Dengan Python. *Jurnal Informatika dan Teknik Elektro Terapan*, 13, 2830-7062.

Raup, A., Ridwan, W., Khoeriyah, Y., Supiana, & Zaqiah, Q. Y. (2022). Deep Learning dan Penerapannya dalam Pembelajaran. *Jurnal Ilmiah Ilmu Pendidikan*, 5, 3258-3267.