**Laporan Praktikum Minggu Ke-2 Kontrol Cerdas**

**Minggu ke-2**

Nama : Ilham Bintang Sebastian

NIM : 224308058

Kelas : TKA 7C

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

**1. Judul Percobaan**

Object Classification with Scikit-learn

**2. Tujuan Percobaan**

Tujuan dari percobaan pada praktikum kali ini, sebagai berikut:

1. Memahami konsep dasar kontrol cerdas (intelligent control systems).
2. Mengintegrasikan model ML dengan Computer Vision untuk deteksi objek.
3. Mengelola dataset dan melakukan pelatihan model sederhana.

**3. Landasan Teori**

### Machine Learning dalam Sistem Kendali

Machine Learning (ML) Pembelajaran Mesin (ML) merupakan bagian dari kecerdasan buatan yang memungkinkan sistem untuk memperoleh pengetahuan dari data serta meningkatkan kinerjanya tanpa perlu pemrograman yang jelas. Dalam konteks pengendalian, ML berfungsi untuk mengatasi batasan dari metode tradisional yang umumnya bergantung pada model matematika. Dengan kemampuan dalam mengenali pola dan beradaptasi, ML dapat diterapkan di berbagai jenis sistem kendali seperti pengendalian prediktif, pengendalian adaptif, dan pembelajaran penguatan. Integrasi ini sangat bermanfaat untuk menangani sistem yang kompleks, nonlinier, dan sulit diuraikan, sehingga pengendalian bisa menjadi lebih efisien dan responsif. Namun, penerapan ML di bidang pengendalian juga menghadapi berbagai tantangan, seperti perlunya data yang melimpah, kesulitan dalam interpretasi hasil, serta kebutuhan daya komputasi yang tinggi. Meskipun demikian, kombinasi antara sistem kendali dan ML semakin mendapati relevansinya dalam bidang robotika, otomasi industri, dan kendaraan mandiri.

### Algoritma K-Nearest Neighbors (KNN)

Algoritma K-Nearest Neighbors (KNN) merupakan salah satu metode sederhana namun cukup efektif dalam *supervised learning*. Konsep utamanya adalah menentukan kelas suatu data dengan melihat mayoritas kelas dari *k* tetangga terdekat pada ruang fitur. Proses KNN meliputi: menghitung jarak antara data uji dan seluruh data latih, misalnya menggunakan *Euclidean Distance*, menentukan nilai *k* sebagai jumlah tetangga terdekat, dan menetapkan kelas mayoritas dari tetangga tersebut sebagai hasil prediksi. KNN banyak digunakan dalam klasifikasi berbasis warna karena mudah diterapkan dan memberikan akurasi yang baik pada dataset berukuran kecil

### Computer Vision dengan OpenCV

OpenCV (Open Source Computer Vision Library) adalah pustaka pemrograman untuk pemrosesan citra dan video. Pada praktikum ini, OpenCV digunakan untuk menangkap citra dari kamera, membaca nilai RGB pada piksel tertentu seperti bagian tengah gambar, serta menampilkan hasil klasifikasi warna secara real-time di layar.

### Preprocessing Data

Supaya model Machine Learning dapat berfungsi secara optimal, data harus melalui tahap *preprocessing* terlebih dahulu. Pada praktikum ini, langkah yang dilakukan meliputi normalisasi data menggunakan StandardScaler untuk menyamakan skala antar fitur RGB, serta membagi dataset menjadi data latih (training set) dan data uji (testing set) agar kinerja model dapat dievaluasi dengan lebih objektif.

### Version Control dengan GitHub

GitHub digunakan sebagai media untuk menyimpan, mengelola, dan membagikan kode hasil praktikum. Dengan version control, setiap perubahan dapat dilacak sehingga memudahkan kolaborasi dan dokumentasi

**4. Analisis dan Diskusi**

**Analisis**

1. Bagaimana performa model dalam mendeteksi warna?

Model Machine Learning (ML) dapat digunakan untuk identifikasi warna pada citra digital. Algoritma pembelajaran mendalam dapat digunakan untuk mengekstrak fitur warna dari gambar dan membuat model warna. Dimana terdapat poin pada ouput yang dihasilkan yaitu akurasi. Tingkat akurasi identifikasi warna bisa mencapai sekitar 86.8%, yang dipengaruhi oleh resolusi kamera, kualitas lensa, dan pencahayaan. Algoritma K-Nearest Neighbor (KNN) dapat digunakan dalam klasifikasi warna dengan tingkat akurasi yang tinggi, lebih dari 90%.

2. Bagaimana perbedaan akurasi jika jumlah dataset ditambah?

Penambahan jumlah dataset pada metode machine learning dapat meningkatkan tingkat akurasi model, dapat lebih baik dalam mengklasifikasi data dan dapat meningkatkan kemampuan dalam mendeteksi suatu objek.

3. Bagaimana cara meningkatkan kinerja model klasifikasi?

Untuk meningkatkan kinerja model klasifikasi dapat dilakukan dengan cara:

a) Mengoptimasi model yang ingin digunakan.

b) Meningkatkan kualitas data hingga dataset yang digunakan memiliki jumlah sample yang cukup.

c) Memilih fitur yang lebih relevan dengan model yang ingin dibuat.

**Diskusi**

1. Apa keuntungan Machine Learning dibandingkan metode berbasis aturan (rule-based)?

Machine Learning (ML) memiliki beberapa keuntungan dibandingkan metode berbasis aturan (rule-based)yang dapat dilihat dari:

a) Identifikasi Pola dan Tren: ML dapat mengidentifikasi pola dan tren data yang mungkin terlewatkan oleh manusia. Teknologi ML juga digunakan untuk mengolah data berukuran besar, termasuk menemukan pola dan tren big data dalam waktu relatif singkat.

b) Akurasi Meningkat Seiring Waktu: ML terus belajar berdasarkan data yang ada, sehingga akurasi hasil prediksi terus meningkat seiring banyaknya data yang dipelajari.

c) Pengambilan Keputusan Otomatis: ML dapat digunakan untuk mengotomatisasi proses pengambilan keputusan, menghemat waktu dan sumber daya manusia.

d) Kemampuan Generalisasi: Model ML hanya dapat memberikan hasil yang baik pada data yang serupa dengan data pelatihan.

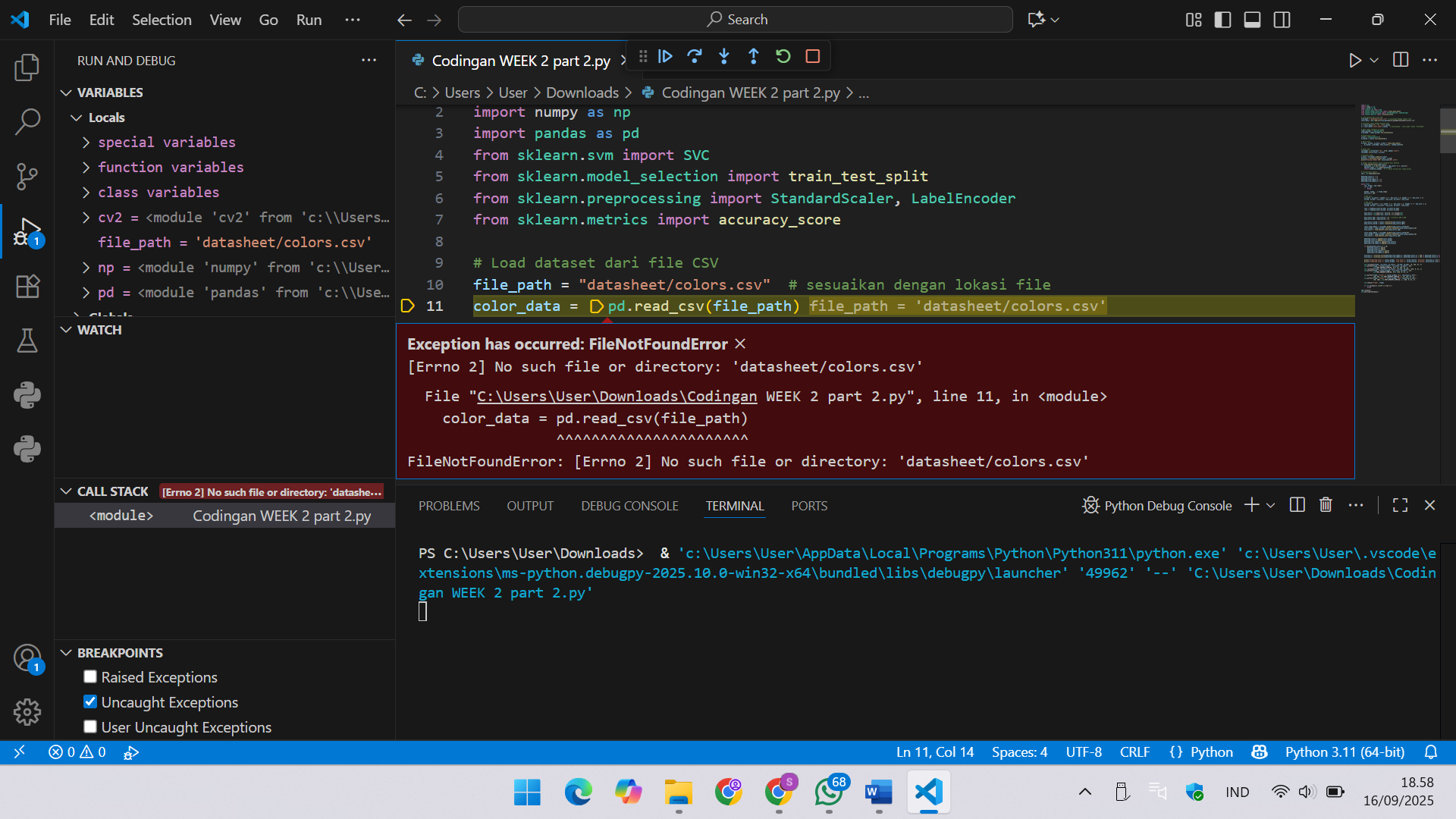
e) Tanpa Intervensi Manusia: Model ML dapat bekerja tanpa intervensi manusia setelah disiapkan.

f) Mendeteksi Anomali: Software pada ML dapat digunakan untuk menyortir data dengan kriteria poin data, pola data, dan juga anomali data-data yang bersifat penting.

2. Bagaimana ML dapat diintegrasikan lebih lanjut dalam sistem kendali?

Machine Learning (ML) dapat diintegrasikan lebih lanjut dalam sistem kendali untuk menciptakan sistem yang lebih cerdas dan adaptif. Misalnya, dalam robotika, ML dapat digunakan untuk mengenali objek dan menyesuaikan gerakan secara otomatis. Dalam sistem kendali industri, ML bisa menganalisis data sensor secara real-time untuk mengoptimalkan performa mesin, mengurangi konsumsi energi, dan mencegah kerusakan dengan deteksi dini. Selain itu, pada kendaraan otonom, ML memungkinkan sistem untuk mengenali rintangan, mengatur kecepatan, dan membuat keputusan berkendara yang lebih aman. Sehingga dengan terus belajar dari data, sistem kendali berbasis ML menjadi lebih akurat, efisien, dan mampu beradaptasi dengan perubahan lingkungan.

**Kendala dan Solusi**



Pada saat menjalankan program, muncul error **FileNotFoundError: [Errno 2] No such file or directory: 'colors.csv'.** Hal ini terjadi karena program tidak menemukan file dataset yang dimaksud. Secara struktur folder, file colors.csv sebenarnya tersimpan di dalam subfolder **datasheet,** sedangkan pada kode pemanggilan masih ditulis 'colors.csv' seolah-olah berada di folder utama. Akibatnya, Python gagal membaca file tersebut. Kendala ini dapat diatasi dengan menyesuaikan path file sesuai lokasi penyimpanan.

1. **Assignment**

Awalnya, sistem hanya mampu mendeteksi satu warna dengan model machine learning sederhana. Setelah dimodifikasi, program dapat mengenali beberapa warna sekaligus melalui penambahan area deteksi (ROI) dan penyesuaian algoritma klasifikasi berbasis OpenCV serta SVM/KNN dengan fitur RGB/HSV. Hasil deteksi ditampilkan di layar menggunakan bounding box dan label teks berisi nama warna serta akurasinya. Untuk meningkatkan ketepatan, sistem juga dilengkapi datasheet warna yang lebih lengkap, mencakup warna dasar maupun variasinya seperti abu-abu, cokelat, dan ungu.

1. **Data dan Output Hasil Pengamatan**

Setelah dilakukan pengujian pada program yang telah dimodifikasi dengan menerapkan metode Support Vector Machine (SVM), sistem mampu mendeteksi warna objek yang ditangkap kamera secara real-time. Setiap objek terdeteksi diberi bounding box dan label warna sesuai hasil prediksi, serta ditampilkan pula nilai akurasi deteksinya pada layar. Hasil pengujian menunjukkan bahwa sistem dapat mengenali warna dengan cukup baik, meskipun tingkat akurasi masih dipengaruhi oleh kondisi pencahayaan dan kesamaan warna dengan latar belakang, sebagaimana terlihat pada tabel hasil pengamatan berikut.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Warna Objek** | **Hasil Deteksi** | **Keterangan** | **Foto** |
| 1 | Biru | Terdeteksi | Bounding box biru dengan label Biru |  |
| 2 | Kuning | Terdeteksi | Bounding box kuning dengan label kuning terdeteksi benar, |  |

**7. Kesimpulan**

Berdasarkan hasil praktikum, dapat disimpulkan bahwa penerapan metode Machine Learning Support Vector Machine (SVM) yang diintegrasikan dengan Computer Vision melalui OpenCV mampu melakukan deteksi warna objek secara real-time dengan kinerja yang cukup optimal. Sistem yang dikembangkan tidak hanya dapat mengenali lebih dari satu warna secara bersamaan, tetapi juga menampilkan hasil deteksi dalam bentuk bounding box dan label teks yang disertai dengan nilai akurasi. Temuan ini menunjukkan bahwa metode klasifikasi warna berbasis SVM efektif, relatif mudah diimplementasikan, serta sesuai dengan tujuan praktikum. Namun demikian, tingkat akurasi deteksi masih dipengaruhi oleh faktor eksternal, seperti kondisi pencahayaan dan kemiripan warna objek dengan latar belakang.

**8. Saran**

Walaupun sistem deteksi warna yang dikembangkan sudah memberikan hasil yang cukup baik, masih terdapat beberapa aspek yang bisa ditingkatkan untuk memperoleh kinerja yang lebih optimal. Adapun beberapa saran pengembangan yang dapat dilakukan antara lain:

1. Menyesuaikan rentang HSV secara lebih akurat sehingga hasil deteksi tidak terlalu dipengaruhi oleh perubahan pencahayaan.
2. Mengintegrasikan deteksi warna dengan metode lain, seperti deteksi bentuk atau pola, guna meningkatkan akurasi identifikasi objek.
3. Mengembangkan sistem dengan memanfaatkan algoritma AI/Deep Learning sehingga lebih cerdas, adaptif, dan mampu mengatasi situasi nyata yang lebih kompleks.