

Laporan Praktikum 2 Kontrol Cerdas

Nama : Atniko Dwi Saputra

NIM : 224308005

Kelas : TKA – 6A

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

Student Lab Assistant : Muhammad Mahirul Faiq / 214308043

1. Judul Percobaan

Judul : Deteksi Klasifikasi Objek Dengan Fitur Warna RGB

2. Tujuan Percobaan

Pada praktikum ini, mahasiswa akan dapat:

- Memahami dasar-dasar Machine Learning dalam sistem kendali.
- Mengimplementasikan model ML sederhana untuk klasifikasi objek.
- Menggunakan Scikit-learn untuk membuat model ML dasar.
- Mengintegrasikan model ML dengan Computer Vision untuk deteksi objek.
- Mengelola dataset dan melakukan pelatihan model sederhana.

3. Landasan Teori

Machine Learning (ML) adalah cabang dari kecerdasan buatan (Artificial Intelligence) yang memungkinkan sistem komputer untuk belajar dari data dan meningkatkan kinerjanya tanpa diprogram secara eksplisit. ML bekerja dengan menganalisis pola dalam data historis untuk membuat prediksi atau keputusan yang lebih akurat di masa depan. Secara umum, ML terbagi menjadi tiga kategori utama: Supervised Learning, yang menggunakan data berlabel untuk membangun model prediktif; Unsupervised Learning, yang mengidentifikasi pola dalam data tanpa label eksplisit; dan Reinforcement Learning, yang mengoptimalkan tindakan melalui interaksi dengan lingkungan berbasis sistem reward dan penalti. Algoritma ML dapat berupa Decision Trees, Support Vector Machines (SVM), Neural Networks, atau Deep Learning, yang digunakan dalam berbagai bidang seperti pengenalan pola, analisis prediktif, otomasi sistem kendali, dan pemrosesan bahasa alami. Dalam konteks sistem kendali, ML digunakan untuk meningkatkan adaptabilitas dan efisiensi sistem dengan mengatasi keterbatasan model berbasis aturan atau matematis konvensional, sehingga menghasilkan solusi yang lebih fleksibel dan presisi dalam berbagai kondisi dinamis.

4. Analisis dan Diskusi

Analisis :

1. Bagaimana performa model dalam mendeteksi warna?

- Model berbasis *Support Vector Machine (SVM)* yang digunakan memiliki akurasi yang cukup

tinggi dalam mendeteksi warna berdasarkan dataset latihannya.

- Dari hasil pengujian, model dapat mengenali warna hitam, putih, dan warna lain dengan akurat,

2. Bagaimana perbedaan akurasi jika jumlah dataset ditambah?

- Meningkatkan jumlah dataset umumnya akan meningkatkan akurasi model, karena model memiliki lebih banyak variasi data untuk dipelajari.
- Dengan lebih banyak sampel warna yang mencakup variasi pencahayaan, sudut pandang, dan kondisi lingkungan yang berbeda, model dapat:
 - Lebih baik dalam mengenali warna dengan rentang nilai yang lebih luas.
 - Mengurangi kesalahan klasifikasi akibat overfitting pada dataset yang kecil.

3. Bagaimana cara meningkatkan kinerja model klasifikasi?

- Mengoptimalkan rentang warna HSV:
 - Menyesuaikan nilai *Hue*, *Saturation*, dan *Value* untuk menangkap variasi warna yang lebih baik.
 - Menggunakan metode *adaptive thresholding* untuk membuat deteksi lebih robust terhadap perubahan pencahayaan.
- Menggunakan model ML yang lebih kompleks:
 - Misalnya, menerapkan Convolutional Neural Networks (CNN) untuk meningkatkan klasifikasi warna berdasarkan fitur spasial dalam gambar.
- Preprocessing gambar sebelum klasifikasi:
 - Menerapkan teknik white balance correction untuk mengurangi efek pencahayaan yang tidak merata.
 - Menggunakan Gaussian Blur atau Median Filtering untuk mengurangi noise sebelum ekstraksi fitur warna.
- Menambahkan lebih banyak data latih yang beragam:
 - Menggunakan augmentasi data, seperti perubahan kecerahan atau kontras, untuk meningkatkan generalisasi model.
 - Mengumpulkan lebih banyak sampel warna dalam berbagai kondisi pencahayaan dan latar belakang.
- Menggunakan teknik kalibrasi warna:
 - Menggunakan kartu referensi warna (misalnya *Macbeth ColorChecker*) untuk memastikan warna yang diambil oleh kamera lebih akurat.

Diskusi :

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

- Metode berbasis aturan (*rule-based systems*) memerlukan aturan eksplisit yang harus didefinisikan secara manual, sedangkan *Machine Learning (ML)* dapat belajar dari data tanpa perlu pemrograman eksplisit untuk setiap kondisi.
- ML dapat terus diperbarui dengan data baru, sedangkan rule-based perlu disesuaikan secara manual jika terjadi perubahan lingkungan atau kondisi pencahayaan.
- ML lebih efektif dalam menangani dataset dengan kompleksitas tinggi dan non-linearitas yang sulit direpresentasikan menggunakan aturan eksplisit.

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

- Dalam kasus deteksi warna, ML dapat menggantikan pendekatan berbasis *thresholding* HSV dengan model CNN atau SVM untuk mendeteksi warna lebih akurat.
- ML dapat digunakan untuk mendeteksi kesalahan pada sistem kendali secara real-time.
- ML dapat digunakan untuk memprediksi perilaku sistem berdasarkan data historis.

3. Apa saja tantangan dalam penerapan ML dalam sistem real-time?

- Model ML dapat rentan terhadap gangguan eksternal seperti perubahan kondisi lingkungan atau data sensor yang tidak akurat.
- Model ML yang kurang data bisa mengalami *underfitting*, sementara terlalu banyak data bisa

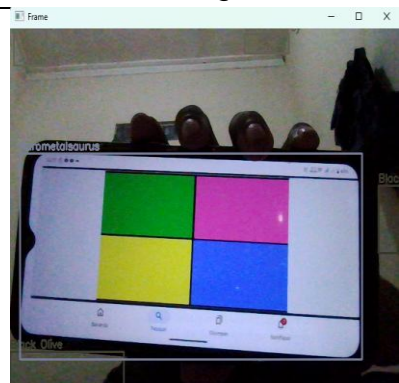
menyebabkan *overfitting* dan meningkatkan kebutuhan daya komputasi.

5. Assignment

- Program coding SVM menggunakan colors.csv
Kode ini menerapkan deteksi warna berbasis *machine learning* menggunakan algoritma Support Vector Machine (SVM) untuk mengklasifikasikan warna berdasarkan nilai RGB yang diambil dari dataset CSV. Prosesnya diawali dengan *preprocessing* data, termasuk normalisasi serta pembagian dataset menjadi data latih dan data uji. Model SVM dengan kernel linear kemudian dilatih, dievaluasi, dan disimpan menggunakan joblib, sehingga dapat digunakan kembali tanpa perlu pelatihan ulang. Dalam implementasi *real-time*, sistem memanfaatkan OpenCV untuk menangkap gambar dari kamera, mengonversinya ke format HSV, serta mendeteksi warna menggunakan fungsi cv2.inRange dan cv2.findContours. Akurasi deteksi sangat bergantung pada kualitas dataset dan kondisi pencahayaan, sehingga mungkin diperlukan teknik pengolahan citra tambahan untuk meningkatkan keandalan. Secara keseluruhan, kombinasi machine learning dan computer vision ini memiliki berbagai aplikasi, seperti di bidang industri, desain, navigasi robot, dan teknologi *augmented reality*.
- Program coding SVM menggunakan training_datasheet
Kode ini menerapkan deteksi warna berbasis *machine learning* menggunakan Support Vector Machine (SVM) dengan dataset berupa gambar warna yang tersimpan dalam folder terstruktur. Setiap gambar diproses menggunakan OpenCV, dikonversi ke format RGB, dan dihitung nilai rata-rata (R, G, B) sebagai fitur untuk pelatihan model. Selanjutnya, data dinormalisasi menggunakan StandardScaler dan dibagi menjadi data latih serta data uji sebelum model SVM dengan kernel linear dilatih dan dievaluasi. Model yang telah terlatih kemudian disimpan menggunakan joblib, memungkinkan penggunaan kembali tanpa perlu pelatihan ulang untuk keperluan deteksi warna *real-time* melalui kamera. Dalam implementasi *real-time*, video dari kamera dikonversi ke format HSV, lalu dilakukan segmentasi menggunakan cv2.inRange untuk mendeteksi objek berdasarkan warna. Kontur yang terdeteksi dianalisis, nilai rata-rata warna dihitung, dan diklasifikasikan menggunakan model SVM, dengan hasil ditampilkan dalam bentuk bounding box serta label warna. Sistem ini memiliki berbagai aplikasi, seperti pengenalan warna otomatis, analisis gambar, serta navigasi berbasis warna, yang dapat digunakan di berbagai industri dan teknologi.

6. Data dan Output Hasil Pengamatan

Sajikan data dan hasil yang diperoleh selama percobaan. Gunakan tabel untuk menyajikan data jika diperlukan.

No	Variabel	Hasil Pengamatan
1	KNN colors.csv	

2	SVM colors.csv	
3	SVM training_datasheet	

7. Kesimpulan

Program 1 menggunakan colors.csv, yang merupakan kumpulan data berbasis tabel dengan nilai RGB yang terstruktur dan seragam, sedangkan Program 2 menggunakan training_dataset, yang terdiri dari kumpulan gambar yang dikonversi menjadi fitur berdasarkan rata-rata warna. Dalam hal akurasi model, Program 1 cenderung memiliki akurasi lebih tinggi untuk warna yang terstruktur karena nilai RGB berasal dari dataset yang lebih bersih dan terkontrol. Sementara itu, Program 2 lebih fleksibel dalam mendeteksi warna dari gambar dunia nyata, tetapi akurasinya bisa lebih rendah jika dataset training_dataset memiliki pencahayaan yang tidak seragam atau jumlah sampel warna yang tidak seimbang..

8. Saran

Jika menginginkan akurasi tinggi dalam kondisi pencahayaan yang stabil, Program 1 dengan colors.csv adalah pilihan yang tepat. Namun, agar model dapat mengenali lebih banyak warna, dataset perlu mencakup lebih banyak variasi warna. Jika membutuhkan fleksibilitas dalam mendeteksi warna di berbagai kondisi dunia nyata, Program 2 lebih cocok, tetapi kualitas dataset training_dataset harus ditingkatkan dengan gambar yang lebih jelas dan pencahayaan yang lebih merata. Untuk hasil yang lebih optimal, kedua pendekatan dapat digabungkan dengan menambahkan dataset colors.csv sebagai referensi dalam Program 2, sehingga model dapat mengenali warna dengan lebih akurat..

9. Daftar Pustaka

- Alpaydin, E. (2020). *Introduction to Machine Learning (4th ed.)*. MIT Press.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.)*. Springer.