

Laporan Praktikum Kontrol Cerdas

Nama : Moh.Farhan Baihaqi

NIM : 224308037

Kelas : TKA-6B

Akun Github (Tautan) : <https://github.com/farhanbaihaqi21>

Student Lab Assistant :Mbak Dwi

1. Judul Percobaan : ***Machine Learning for Control Systems***

2. Tujuan Percobaan :

- Memahami konsep dasar *Machine Learning for Control Systems*.
- Mengimplementasikan model ML sederhana untuk klasifikasi objek.
- Mengelola dataset dan melakukan pelatihan model sederhana.
- Mengintegrasikan model ML dengan Computer Vision untuk deteksi objek.

3. Landasan Teori :

Machine Learning merupakan sistem yang mampu belajar sendiri untuk memutuskan sesuatu tanpa harus berulang kali diprogram oleh manusia (*Karimah Tauhid, Volume 2 Nomor 1 (2023), e-ISSN 2963-590X, 2023*). Untuk bisa menggunakan *machine learning* harus memiliki data. Data yang digunakan dibagi menjadi dua data training dan data testing. Data training digunakan untuk melatih algoritma, sedangkan data testing digunakan untuk mengetahui performa yang telah dilatih sebelumnya ketika menemukan data baru yang belum pernah diberikan dalam data training (Yudianto & Fatta, 2020). *Machine Learning* memiliki beberapa jenis, seperti, *supervised learning*, *unsupervised learning* dan *reinforcement learning*.

Supervised learning merupakan salah satu teknik machine learning yang menggunakan dataset (*data training*) yang sudah berlabel (*labeled data*) untuk melakukan pembelajaran pada mesin, sehingga mesin mampu mengidentifikasi label input dengan menggunakan fitur yang dimiliki untuk selanjutnya melakukan prediksi maupun klasifikasi. Klasifikasi yaitu proses mengelompokkan objek-objek

dengan karakteristik yang mirip dalam beberapa kelas. Pada umumnya pengklasifikasian dokumen diwakili oleh kalimat-kalimat penting dengan menentukan ciri-ciri atau karakteristik (Septhya et al., 2023). Ada beberapa klasifikasi dalam *supervised learning* diantaranya *Decision Tree*, *K-Nearest Neighbor (KNN)*, *Naive Bayes*, *Regresi*, dan *Super Vector Machine* (Retnoningsih & Pramudita, 2020). Salah satu metode dalam klasifikasi adalah *Super Vector Machine*. *Super Vector Machine* berfungsi untuk memperoleh hasil prediksi pengujian, yang mana hasil dari prediksi untuk pengujian diperoleh dari kelompok yang berupa *feature vector*. *Support Vector Machine (SVM)* adalah teknik seleksi yang menghasilkan hasil dengan tingkat akurasi klasifikasi tertinggi dengan membandingkan sekumpulan parameter standar dengan nilai diskrit yang disebut sebagai himpunan kandidat. Kelebihan dari *Support Vector Machine (SVM)* selain dari populer juga sangat cocok untuk klasifikasi dikarenakan tidak bergantung pada jumlah atribut dan dapat menyelesaikan masalah dari dimensi (Septhya et al., 2023).

4. Analisis dan Diskusi :

-Analisis

Percobaan ini melakukan deteksi warna objek menggunakan klasifikasi model SVM (*Support Vector Machine*). Model SVM mengklasifikasi warna berdasarkan nilai RGB yang diambil dari kamera, kemudian mencari **hyperplane** terbaik dalam mengklarifikasinya. Pada program SVM menggunakan kernel RBF (Radial Basis Fuction) untuk menangkap hubungan non-linear antar fitur, selanjutnya $C=10$ memberikan margin untuk pemisahan kelas dan $\text{Gamma} = \text{scale}$ untuk menghitung jarak antar titik data. Untuk proses klasifikasi warna dengan SVM dengan membaca dataset dari file `colors2.csv` yang berisi nilai R, G, B dan ColorName. Kemudian melakukan normalisasi menggunakan **StandardScaler()** agar setiap fitur memiliki skala yang sama, kemudian melakukan pelatihan model SVM menggunakan **`svm_model.fit(X_train, y_train)`** dimana nilai RGB yang telah dinormalisasi digunakan sebagai fitur dan label warna sebagai target, model akan mencari

pola dalam data sehingga dapat mengenali warna berdasarkan nilai RGB. Selanjutnya model SVM diuji menggunakan `svm_model.predict(X_test)` dan akurasi dihitung dengan `accuracy_score(y_test, y_pred)`, akurasi model memberikan gambaran tentang seberapa baik model dapat mengenali warna dengan benar. Kemudian mengimplementasikan deteksi warna dari kamera menggunakan OpenCV untuk menampilkan gambar secara real-time dan terdapat tampilan 2 bounding box untuk mendeteksi 2 warna berbeda.

-Diskusi

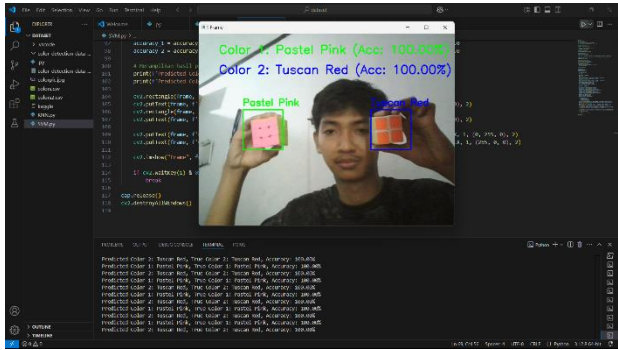
Program ini mengklasifikasikan warna berdasarkan nilai RGB menggunakan model SVM. Untuk membangun model SVM untuk deteksi objek dengan melakukan beberapa Langkah yaitu: Mempersiapkan dataset Warna, Pra-pemrosesan data dengan mengkonversi label ke dalam bentuk Numerik, Normalisasi data, split data untuk Training dan Testing, membangun model SVM dengan kernel RBF untuk membuat data menjadi non-linear, evaluasi model untuk mengukur akurasi pada data uji, implementasi deteksi warna dari kamera menggunakan openCV


5. Assignment :

Pada program dilakukan dengan mengimport beberapa *library* seperti : cv2 untuk pemrosesan video, numpy untuk manipulasi array, pandas digunakan untuk membaca file dari CSV, untuk proses klasifikasi menggunakan model SVM dengan mengimport sklearn. Selanjutnya memproses program untuk pelatihan model dengan memilih variabel x dan y dari dataset, variabel x untuk (R,G,B) dan variabel y (ColorName). Kemudian membuat label kelas, label kelas pada SVM bekerja dengan data numerik sehingga diubah menjadi angka menggunakan LabelEncoder dengan library : **from sklearn.preprocessing import LabelEncoder** untuk programnya `label_encoder = LabelEncoder()` `y_encoded = label_encoder.fit_transform(y)`. Kemudian dengan menormalisasi data agar setiap variabel memiliki skala yang sama, dengan library : **from sklearn.preprocessing import StandardScaler** untuk programnya `scaler = StandardScaler()` `X_scaled =`

scaler.fit_transform(X). Selanjutnya, dataset dilakukan training set dan testing set agar program bisa diuji dengan data baru dengan library **from sklearn.model_selection import train_test_split**, untuk programnya **X_train, X_test, y_train, y_test = train_test_split(X_scaled, y_encoded, test_size=0.2, random_state=42)**. Klasifikasi SVM bisa dilatih dengan dataset yang sudah diproses dengan inisialisasi model SVM seperti, menentukan atau memisahkan data menggunakan kernel, menentukan **regularization parameter** untuk mengontrol keseimbangan antara akurasi dan kelengkapan model, menggunakan gamma dalam kernel non-linear seperti RBF(**Radial Basis Function**) dengan library **from sklearn.svm import SVC** untuk programnya **svm_model = SVC(kernel='rbf', C=10, gamma='scale')** **svm_model.fit(X_train, y_train)** Setelah klasifikasi model SVM dilatih, selanjutnya mengukur akurasi dengan data uji dengan library **from sklearn.metrics import accuracy_score** untuk programnya **y_pred = svm_model.predict(X_test)** **accuracy = accuracy_score(y_test, y_pred)** **print(f"Akurasi Model SVM: {accuracy*100:.2f}%")**. Kemudian mengimplementasikan model SVM kedalam kamera untuk deteksi warna. Di dalam tampilan warna terdapat 2 bounding box yang digunakan untuk membatasi 2 objek yang akan dideteksi dan dalam tampilan kamera terdapat nilai akurasi dari masing" warna yang dideteksi.

6. Data dan Output Hasil :

No.	Variabel	Hasil Pengamatan
1.	Akurasi Model SVM	

2.	<p><i>Running Accuracy Real-Time</i></p>	
----	--	--

7. Kesimpulan :

Berdasarkan praktikum dan analisis yang telah dilakukan, maka dapat diambil kesimpulan yaitu:

- Program ini dibuat untuk mendeteksi dan mengklasifikasi warna secara real-time menggunakan model SVM.
- Program ini menggunakan machine learning dengan jenis supervised learning yang menggunakan dataset (data training) yang sudah berlabel (labeled data) untuk melakukan proses programnya.
- Model SVM dengan kernel RBF menggunakan data training dan testing untuk menghitung akurasi.
- Hasil dari program menampilkan gambar dan dua bounding box untuk mengambil sampel warna, hasil sampel warna akan ditampilkan secara real-time.

8. Saran :

Untuk meningkatkan akurasi deteksi dengan mengurangi noise dari kamera agar akurasi meningkat saat mendeteksi warna objek, Kemudian bisa mengubah RGB ke dalam HSV karena saat mendeteksi faktor yang mempengaruhi akurasi dalam mendeteksi warna adalah tingkat pencahayaan dari lingkungan sekitar.

9. Daftar Pustaka :

Karimah Tauhid, Volume 2 Nomor 1 (2023), e-ISSN 2963-590X. (2023). 2.

Retnoningsih, E., & Pramudita, R. (2020). Mengenal Machine Learning Dengan Teknik Supervised Dan Unsupervised Learning Menggunakan Python. *BINA INSANI ICT JOURNAL*, 7(2), 156. <https://doi.org/10.51211/biict.v7i2.1422>

Septhya, D., Rahayu, K., Rabbani, S., Fitria, V., Rahmadden, R., Irawan, Y., & Hayami, R. (2023). Implementasi Algoritma Decision Tree dan Support Vector Machine untuk Klasifikasi Penyakit Kanker Paru: Implementation of Decision Tree Algorithm and Support Vector Machine for Lung Cancer Classification. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 3(1), 15–19. <https://doi.org/10.57152/malcom.v3i1.591>

Yudianto, M. R. A., & Fatta, H. A. (2020). *ANALISIS PENGARUH TINGKAT AKURASI KLASIFIKASI CITRA WAYANG DENGAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK.*