

WEEK 3

Laporan Praktikum Kontrol Cerdas

Nama : Vivi Aulia Husna Wahidah
NIM : 224308023
Kelas : TKA-6A
Akun Github (Tautan) : <https://github.com/vivi-aulia>
Student Lab Assistant : Muhammad Mahirul Faiq

1. Judul Percobaan

Week 3: Deep Learning for Intelligent Control Systems

2. Tujuan Percobaan

Tujuan dari praktikum “Object Classification with Machine”, mahasiswa diharapkan mampu:

1. Memahami konsep dasar Deep Learning dalam sistem kendali.
2. Mengimplementasikan Convolutional Neural Network (CNN) untuk klasifikasi objek.
3. Menggunakan TensorFlow dan Keras untuk membangun model Deep Learning.
4. Mengintegrasikan model CNN dengan Computer Vision untuk deteksi objek secara real-time.
5. Menggunakan dataset dari Kaggle untuk pelatihan model.
6. Mengembangkan mode Night Vision untuk deteksi objek dalam kondisi pencahayaan rendah.

3. Landasan Teori

1. Konsep Dasar Deep Learning dalam Sistem Kendali

Deep Learning merupakan cabang dari pembelajaran mesin yang menggunakan jaringan saraf tiruan dengan banyak lapisan untuk memodelkan data yang kompleks. Dalam sistem kendali, Deep Learning digunakan untuk mengidentifikasi dan memodelkan sistem dinamis berdasarkan data masukan dan keluaran. Pendekatan ini memungkinkan perancangan sistem kendali yang adaptif dan mampu menangani non-linearitas serta ketidakpastian dalam model sistem.

2. Implementasi Convolutional Neural Network (CNN) untuk Klasifikasi Objek

Convolutional Neural Network (CNN) adalah arsitektur jaringan saraf yang dirancang khusus untuk memproses data berbentuk grid, seperti citra. CNN terdiri dari lapisan konvolusi yang berfungsi mengekstraksi fitur-fitur penting dari citra, sehingga sangat efektif untuk tugas klasifikasi objek. Implementasi CNN telah terbukti berhasil dalam berbagai aplikasi, termasuk klasifikasi citra candi, klasifikasi objek umum, dan klasifikasi citra tanah liat.

3. Penggunaan TensorFlow dan Keras untuk Membangun Model Deep Learning

TensorFlow dan Keras adalah pustaka open-source yang populer untuk membangun dan melatih model Deep Learning. TensorFlow menyediakan infrastruktur untuk komputasi numerik yang efisien, sementara Keras menawarkan antarmuka tingkat tinggi yang memudahkan perancangan dan eksperimen dengan berbagai arsitektur jaringan saraf.

4. Integrasi Model CNN dengan Computer Vision untuk Deteksi Objek Secara Real-Time

Integrasi model CNN dengan teknik Computer Vision memungkinkan deteksi objek secara real-time, yang penting untuk aplikasi seperti kendaraan otonom dan sistem pengawasan. Pendekatan ini melibatkan penggunaan CNN untuk mengenali dan mengklasifikasi objek dalam aliran video atau citra yang diambil secara langsung.

5. Penggunaan Dataset dari Kaggle untuk Pelatihan Model

Kaggle adalah platform yang menyediakan berbagai dataset untuk keperluan pelatihan dan

evaluasi model pembelajaran mesin. Penggunaan dataset dari Kaggle memungkinkan pengembangan model yang lebih robust dan generalisasi yang lebih baik karena keragaman dan ukuran data yang tersedia.

6. Pengembangan Mode Night Vision untuk Deteksi Objek dalam Kondisi Pencahayaan Rendah

Deteksi objek dalam kondisi pencahayaan rendah memerlukan teknik khusus, seperti pengolahan citra dengan filter khusus atau penggunaan teknologi inframerah. Pengembangan mode night vision dapat meningkatkan kemampuan sistem dalam mengenali objek meskipun dalam kondisi cahaya yang minim.

4. Analisis dan Diskusi

Analisis hasil:

1. Penerapan CNN untuk Klasifikasi Gambar

Program ini menggunakan Convolutional Neural Network (CNN) untuk melakukan klasifikasi gambar menjadi enam kategori (building, forest, glacier, mountain, sea, street). Model CNN telah dilatih sebelumnya menggunakan dataset dari Kaggle, yang memungkinkan sistem untuk melakukan klasifikasi secara real-time berdasarkan input dari kamera.

2. Langkah-langkah utama yang dilakukan dalam klasifikasi:

- Preprocessing Gambar: Ukuran gambar disesuaikan menjadi (150, 150) piksel dan dinormalisasi dalam rentang [0,1] sebelum dimasukkan ke dalam model.
- Prediksi Menggunakan Model CNN: Model mengeluarkan probabilitas kelas, dan kelas dengan probabilitas tertinggi dipilih sebagai hasil klasifikasi.
- Menampilkan Hasil Klasifikasi: Hasil prediksi ditampilkan di layar secara real-time menggunakan OpenCV.

3. Mode Night Vision dan Pengolahan Citra

Fitur Night Vision diterapkan dengan menggunakan teknik pemrosesan citra berikut:

- Konversi ke Skala Abu-abu: Menggunakan `cv2.cvtColor()` untuk mengubah gambar menjadi grayscale.
- Peningkatan Kontras: `cv2.equalizeHist()` diterapkan untuk meningkatkan kontras dalam kondisi pencahayaan rendah.
- Pewarnaan Ulang (Color Mapping): `cv2.applyColorMap(gray, cv2.COLORMAP_JET)`

Langkah ini membantu meningkatkan visibilitas dalam lingkungan dengan pencahayaan rendah, sesuai dengan konsep Night Vision dalam sistem kendali berbasis visi.

4. Evaluasi Tingkat Pencahayaan

Program ini juga memiliki fitur untuk mengukur tingkat pencahayaan berdasarkan rata-rata intensitas piksel dalam gambar skala abu-abu. Tingkat pencahayaan diklasifikasikan sebagai berikut:

- **Gelap: Jika nilai rata-rata < 30.**
- **Redup: Jika nilai berada dalam rentang 30 - 100.**
- **Terang: Jika nilai > 100.**

Fitur ini berguna dalam menganalisis kondisi lingkungan secara real-time dan dapat menjadi dasar untuk sistem adaptif dalam kendali berbasis visi.

Diskusi Hasil:

1. Relevansi dengan Deep Learning dalam Sistem Kendali

Program ini mencerminkan aplikasi Deep Learning dalam sistem kendali, seperti yang dijelaskan dalam modul. CNN digunakan dalam berbagai sistem kendali berbasis visi, termasuk:

- Autonomous Vehicles: Untuk deteksi objek dan navigasi otomatis.
- Quality Control: Mendeteksi kecacatan produk dalam industri manufaktur.
- Smart Surveillance: Digunakan dalam sistem keamanan berbasis pengenalan objek.
- Implementasi Night Vision juga menunjukkan bagaimana pemrosesan citra dapat

meningkatkan kinerja sistem dalam kondisi pencahayaan rendah.

2. Keunggulan dan Tantangan dalam Implementasi

Keunggulan:

- Mampu melakukan klasifikasi objek secara real-time.
- Adaptif terhadap kondisi pencahayaan yang berubah-ubah dengan fitur Night Vision.
- Menggunakan metode pemrosesan citra yang efisien untuk meningkatkan visibilitas dalam kondisi minim cahaya.

3. Tantangan:

- Akurasi Model: Prediksi dapat dipengaruhi oleh kualitas dataset pelatihan.
- Performa Real-time: Mungkin mengalami keterlambatan pada perangkat keras dengan spesifikasi rendah.
- Night Vision dengan Histogram Equalization: Kurang efektif dalam menangani noise ekstrem.

4. Pengembangan Lebih Lanjut

Untuk meningkatkan performa dan akurasi sistem, beberapa pengembangan lebih lanjut dapat dilakukan:

1. Peningkatan Akurasi Model:

- Menggunakan transfer learning dengan model seperti VGG16 atau ResNet.
- Memperluas dataset pelatihan dan menerapkan data augmentation.

2. Optimasi Night Vision:

- Menggunakan adaptive histogram equalization untuk peningkatan kontras yang lebih baik.
- Menerapkan Deep Learning untuk Image Enhancement, seperti GAN untuk meningkatkan kualitas gambar dalam kondisi minim cahaya.


3. Integrasi dengan Sistem Kendali:

- Program ini dapat dikembangkan lebih lanjut untuk sistem navigasi berbasis visi dalam robot atau kendaraan otonom.
- Dapat diterapkan dalam sistem pemantauan keamanan berbasis AI untuk deteksi objek dan analisis situasi dalam kondisi minim cahaya.

5. Data dan Output Hasil Pengamatan

Data dan hasil yang diperoleh selama percobaan

No	Variabel	Hasil Pengamatan
1	Kondisi pencahayaan gelap	
2	Kondisi pencahayaan redup	

3	Kondisi pencahayaan terang	
---	----------------------------	---

6. Kesimpulan

Program ini telah berhasil mengimplementasikan CNN untuk klasifikasi gambar serta mode Night Vision guna meningkatkan visibilitas dalam kondisi pencahayaan rendah. Dibandingkan metode berbasis aturan, Machine Learning (ML) lebih fleksibel, dapat beradaptasi, dan mampu menangani data dalam jumlah besar dengan otomatisasi yang mengurangi kesalahan manusia. Integrasi ML dalam sistem kendali dapat diterapkan melalui pengendalian prediktif, adaptive control, serta deteksi dan diagnosis kesalahan yang memungkinkan optimasi proses secara real-time, terutama dengan dukungan IoT dan sensor. Namun, penerapan ML dalam sistem real-time menghadapi tantangan seperti keterbatasan komputasi, latensi, serta ketergantungan pada kualitas data pelatihan. Oleh karena itu, diperlukan optimasi lebih lanjut untuk meningkatkan efisiensi dan keandalan sistem agar dapat diterapkan dalam lingkungan kendali yang lebih kompleks.

7. Saran

1. Optimasi Model dan Efisiensi Sistem

Gunakan transfer learning (VGG16, ResNet) untuk meningkatkan akurasi, serta terapkan quantization dan pruning untuk efisiensi sistem real-time.

2. Peningkatan Night Vision dan Sensor

Tingkatkan visibilitas dengan adaptive histogram equalization atau GAN, dan integrasikan sensor inframerah untuk deteksi lebih akurat di kondisi gelap.

3. Integrasi IoT dan Pengujian Sistem

Hubungkan sistem dengan cloud untuk pemantauan real-time, serta uji model di berbagai kondisi pencahayaan dan perangkat keras dengan simulasi OpenCV atau ROS.

8. Daftar Pustaka

- Pillonetto, G., Aravkin, A., Gedon, D., Ljung, L., Ribeiro, A. H., & Schön, T. B. (2023). Deep networks for system identification: a Survey.
- Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement.
- Singh, K. B., & Arat, M. A. (2019). Deep Learning in the Automotive Industry: Recent Advances and Application Examples.