

Pengembangan Sistem Deteksi Mata Tertutup untuk Pencegahan Kecelakaan di Kendaraan

1. Ide Riset

Kecelakaan lalu lintas disebabkan oleh kelelahan pengemudi menjadi salah satu problem dalam keselamatan berkendara. Untuk mengatasi masalah ini, saya mencoba memikirkan topik **pengembangan sistem deteksi mata tertutup berbasis computer vision**. Penelitian ini bertujuan untuk merancang dan mengembangkan sistem yang mampu mendeteksi mata tertutup sebagai indikator utama untuk pengemudi apakah dalam kondisi mengantuk atau tidak, dan memberikan peringatan dini berupa membunyikan alarm peringatan untuk membangunkan pengemudi dan bisa menghindari kecelakaan yang fatal.

Teknologi ini dirancang untuk diaplikasikan pada kendaraan kecil seperti sepeda motor dan mobil sebagai bagian dari percobaan sistem yang akan saya kembangkan, yang bekerja secara real-time untuk mendeteksi kondisi mata pengemudi menggunakan real-time kamera pada gawai pengguna. Sistem ini memanfaatkan algoritma deteksi wajah, deteksi mata, dan klasifikasi kondisi mata (terbuka atau tertutup) dengan menggunakan **Convolutional Neural Networks (CNN)** atau teknik deep learning yang masih saya cari tahu. Jika sistem mendeteksi mata tertutup selama waktu tertentu (misalnya 5 detik atau lebih), itu akan memberikan peringatan berupa alarm pada gawai, yang ditujukan untuk membangunkan pengemudi atau mengingatkan mereka untuk berhenti dan beristirahat.

2. Metode dan Metodologi

Metode

Metode yang akan digunakan dalam penelitian ini yaitu meliputi deteksi wajah, pelacakan mata, klasifikasi kondisi mata, dan sistem peringatan secara real-time.

1. Deteksi Wajah

Langkah awal dari sistem ini adalah deteksi wajah pengemudi menggunakan algoritma seperti **Haar Cascades** atau model deep learning seperti **Single Shot Multibox Detector (SSD)** yang dikenal efektif dalam mendeteksi objek pada gambar.

Algoritma ini akan mendeteksi posisi wajah pengemudi sebagai dasar untuk mendeteksi mata.

2. Deteksi Mata dan Pelacakan

Setelah wajah terdeteksi, bagian berikutnya dari sistem adalah deteksi mata. **Haar Cascades Classifier** atau **Dlib facial landmarks** digunakan untuk mendeteksi lokasi mata pada wajah. Sistem ini harus dapat bekerja dalam kondisi pencahayaan yang berbeda, sehingga teknik preprocessing seperti **histogram equalization** mungkin diperlukan untuk menormalkan pencahayaan.

3. Klasifikasi Kondisi Mata

Sistem akan mencoba menentukan apakah mata pengemudi terbuka atau tertutup. **Convolutional Neural Networks (CNN)** digunakan untuk melatih model agar dapat mengklasifikasikan gambar mata menjadi dua kategori: terbuka atau tertutup. Dataset dapat digunakan untuk melatih model, dengan melakukan augmentasi data untuk memperluas variasi gambar mata dalam berbagai kondisi pencahayaan dan sudut pandang.

4. Penghitungan Durasi Mata Tertutup

Sistem akan melacak durasi mata tertutup. Jika mata pengemudi tertutup lebih lama dari ambang batas waktu tertentu (misalnya 5 detik), sistem akan mendeteksi ini sebagai tanda kelelahan dan mengeluarkan peringatan.

5. Sistem Peringatan

Sistem peringatan yang diusulkan bisa berbentuk suara, getaran pada kemudi, atau tampilan visual di dashboard kendaraan untuk memperingatkan pengemudi jika mereka mengantuk. Jika pengemudi tidak merespons peringatan, sistem dapat merekomendasikan mereka untuk berhenti dan beristirahat.

Metodologi

Langkah-langkah metodologi berikut yang akan dilakukan dalam pengembangan dan pengujian sistem:

1. Pengumpulan Dataset

Pengumpulan dataset citra wajah pengemudi yang menunjukkan kondisi mata terbuka dan tertutup dari berbagai sudut dan kondisi pencahayaan. Dataset **CEW** (Closed Eyes in the Wild) dan **Driver Monitoring Dataset** yang tersedia secara publik akan digunakan.

2. **Pra-pemrosesan Gambar**

Langkah preprocessing akan dilakukan pada citra untuk meningkatkan kualitas deteksi, termasuk normalisasi pencahayaan, cropping area mata, dan resizing gambar agar sesuai dengan input model CNN.

3. **Pelatihan Model CNN**

Model CNN akan dilatih menggunakan dataset yang sudah diproses untuk mengenali kondisi mata terbuka dan tertutup. **Transfer learning** dapat digunakan untuk meningkatkan akurasi model dan mempercepat pelatihan dengan menggunakan model pre-trained seperti **VGG16** atau **ResNet**.

4. **Pengujian Sistem Real-Time**

Setelah model dilatih, sistem akan diimplementasikan dalam lingkungan real-time seperti didalam mobil atau dipasang pada stanholder motor untuk mendeteksi mata tertutup pada pengemudi. Ini akan diuji dalam berbagai kondisi pencahayaan dan variasi posisi kepala untuk mengukur akurasi deteksi dan ketepatan waktu peringatan.

5. **Evaluasi Kinerja**

Kinerja sistem akan dievaluasi berdasarkan beberapa metrik seperti **precision**, **recall**, **accuracy**, serta **latency** dalam mendeteksi kondisi mata tertutup. Sistem juga akan diuji terhadap berbagai skenario nyata seperti pengemudi yang memakai kacamata hitam atau posisi kepala yang miring.

3. **Referensi**

Berikut beberapa referensi yang dapat mendukung penelitian ini:

1. **Lubis, A. A. R., Purnama, S. I., & Afandi, M. A. (2023). Sistem Pendeteksi Kantuk Berbasis Metode Haar Cascade Untuk Aplikasi Computer Vision. Techno.COM, 22(3), 589-598.**

Penelitian ini mengembangkan sistem untuk mendeteksi kondisi kantuk melalui pengenalan kondisi mata menggunakan metode Haar Cascade, dengan hasil akurasi sebesar 92,5%, precision sebesar 89,71%, dan recall sebesar 96%. Abtahi, S., Omidyeganeh, M., Shirmohammadi, S., & Hariri, B. (2014). Yawning Detection Using Embedded Smart Cameras.

2. **Ramadhani, N., Aulia, S., Suhartono, E., & Hadiyoso, S. (2021). Deteksi Kantuk pada Pengemudi Berdasarkan Penginderaan Wajah Menggunakan PCA dan SVM. *Jurnal Rekayasa Elektronik*, 17(2).**

Penelitian ini memanfaatkan metode PCA dan SVM untuk mendeteksi kantuk pada pengemudi berdasarkan penginderaan wajah, relevan dengan pengembangan sistem deteksi mengantuk. Park, S., Kim, J., Hong, K. S., & Cho, S. (2017). Driver drowsiness detection system based on feature representation learning using various deep networks.

3. **Akbar, Y., Bintoro, B., Mulyana, D. I., & Lestari, S. (2024). Microsleep Detection Using Convolutional Neural Network With ESP32 Sensor on The Driver. *Journal of Information Technology and Computer Science (INTECOMS)*, 7(5), 1668-1674.**

Penelitian ini mengembangkan sistem deteksi microsleep berbasis CNN dengan sensor ESP32, yang relevan dengan pengembangan sistem deteksi kantuk berbasis computer vision

4. **Maharani, U. D., Handayani, A. S., & Lindawati. (2024). Analisis Deteksi Mata Kantuk di Wajah Pengemudi Menggunakan Support Vector Machine (SVM) Berbasis Citra Real-Time. *Building of Informatics, Technology and Science (BITS)*, 6(2), 940-949.**

Artikel ini menjelaskan tentang penggunaan SVM untuk mendeteksi kantuk pada pengemudi, dengan hasil akurasi 85%, yang relevan dengan deteksi kantuk berbasis citra real-time.

5. **Aditya, D. N., Nugraha, C., & Prasetyo, H. (2022). Perancangan Alat Deteksi Dini Kondisi Kantuk untuk Mengurangi Risiko Kecelakaan Kerja Berbasis Pengolahan Citra Digital. *Prosiding Diseminasi FTI*, 1-9.**

Penelitian ini membahas perancangan alat deteksi dini kantuk berbasis Eye Aspect Ratio (EAR) dan Histogram of Oriented Gradient (HOG), dengan implementasi menggunakan Raspberry Pi 3B dan OpenCV.

4. Reference Gap

Meskipun berbagai penelitian telah dilakukan untuk mendeteksi kelelahan pengemudi melalui computer vision, terdapat beberapa **gap** yang dapat dijadikan fokus dalam penelitian ini:

1. **Variasi Pencahayaan:** Pengembangan sistem dapat menggunakan **kamera inframerah** atau sensor tambahan belum dieksplorasi secara menyeluruh.
2. **Penggunaan Kacamata Hitam atau Aksesori Lainnya:** Banyak model deteksi saat ini memiliki kesulitan dalam mendeteksi mata ketika pengemudi menggunakan kacamata, terutama kacamata hitam. Penelitian ini bisa lebih fokus pada pengembangan model yang robust terhadap aksesori seperti kacamata.
3. **Efisiensi Real-Time:** Penggunaan deep learning dalam sistem deteksi real-time sering kali memiliki kendala terkait kecepatan. Mengoptimalkan model agar dapat bekerja dengan cepat dalam kondisi real-time dan pada perangkat keras terbatas.