

IDENTIFIKASI MENGANTUK MENGGUNAKAN ALGORITMA CNN

Adinne Islamiyati¹, Ratih Titi Komalasari²

^{1,2}Informatika, Teknologi Komunikasi Dan Informatika, Universitas Nasional, Indonesia

Article History

Received : Februari 2025
Revised : Maret 2025
Accepted : Maret 2025
Published : Maret 2025

Corresponding author*:

adinneislamiyati07@gmail.com

Cite This Article:

Adinne Islamiyati and Ratih Titi Komalasari, "IDENTIFIKASI MENGANTUK MENGGUNAKAN ALGORITMA CNN", *JUKIM*, vol. 4, no. 2, Mar. 2025.

DOI:

<https://doi.org/10.56127/jukim.v4i2.1946>

Abstract: Fatigue and drowsiness are major factors contributing to traffic accidents and hazardous workplace incidents. Detecting drowsiness in individuals, especially drivers, is crucial for enhancing safety. This study aims to develop a drowsiness detection system using a Convolutional Neural Network (CNN) based on facial images. The CNN algorithm has superior capabilities in recognizing visual patterns, including facial expression changes and eye movement patterns, which are key indicators of drowsiness. This research utilizes a facial image dataset containing open eyes, closed eyes, yawning, and not yawning as drowsiness indicators. The developed CNN model is trained using preprocessing and data augmentation techniques to improve accuracy in detecting drowsy conditions under various lighting conditions and facial angles. The model is evaluated using performance metrics such as accuracy, precision, recall, and F1-score to measure its effectiveness.

The results indicate that the proposed CNN model achieves 96% accuracy on the test dataset in detecting drowsiness. Implementing the model using TensorFlow Lite enables faster and more efficient inference on devices with limited computational resources. This system has the potential to be applied in various applications, such as Driver Monitoring Systems (DMS) or workplace monitoring systems that require user alertness detection.

Keywords: Drowsiness Detection, Convolutional Neural Network (CNN), Deep Learning, Driving Safety, TensorFlow Lite.

Abstrak: Kelelahan dan rasa mengantuk merupakan faktor utama yang berkontribusi terhadap kecelakaan lalu lintas dan insiden kerja yang berbahaya. Deteksi kondisi mengantuk pada individu, terutama pengemudi, menjadi hal yang krusial dalam meningkatkan keselamatan. Penelitian ini bertujuan untuk mengembangkan sistem deteksi mengantuk menggunakan Convolutional Neural Network (CNN) berbasis citra wajah. Algoritma CNN memiliki kemampuan unggul dalam mengenali pola visual, termasuk perubahan ekspresi wajah dan pola mata yang menjadi indikator utama kondisi mengantuk.

Penelitian ini menggunakan dataset citra wajah yang mencakup kondisi mata terbuka, mata tertutup, menguap, dan tidak menguap sebagai indikator mengantuk. Model CNN yang dikembangkan dilatih dengan menggunakan teknik preprocessing dan augmentasi data guna meningkatkan akurasi dalam mendeteksi kondisi mengantuk dalam berbagai situasi pencahayaan dan sudut wajah. Model diuji menggunakan metrik evaluasi seperti akurasi, presisi, recall, dan F1-score untuk mengukur efektivitas deteksi.

Hasil penelitian menunjukkan bahwa model CNN yang dikembangkan mampu mendeteksi kondisi mengantuk dengan akurasi mencapai 96% pada dataset uji. Implementasi model menggunakan TensorFlow Lite juga memungkinkan inferensi berjalan lebih cepat dan efisien pada perangkat dengan spesifikasi terbatas. Sistem ini memiliki potensi untuk diterapkan dalam berbagai aplikasi, seperti Driver Monitoring System (DMS) atau sistem pemantauan kerja yang memerlukan deteksi kewaspadaan pengguna.

Kata Kunci: Deteksi Mengantuk, Convolutional Neural Network (CNN), Deep Learning, Keselamatan Berkendara, TensorFlow Lite.

PENDAHULUAN

Faktor utama penyebab kecelakaan lalu lintas dan berbagai insiden kerja yang berbahaya adalah kelelahan dan rasa mengantuk. Mengantuk saat berkendara menyumbang persentase besar kasus kecelakaan lalu lintas setiap tahun. Mengantuk dapat menyebabkan penurunan respons, gangguan kognitif, dan hilangnya konsentrasi, yang sangat berbahaya dalam situasi yang membutuhkan perhatian tinggi, seperti mengemudi atau mengoperasikan mesin industri.

Untuk mendeteksi kondisi mengantuk, berbagai metode telah dikembangkan, mulai dari yang berbasis fisiologis hingga yang berbasis perilaku. Metode berbasis perilaku lebih praktis karena dapat digunakan menggunakan kamera tanpa perlu berinteraksi langsung dengan tubuh pengguna. CNN telah menghasilkan kemajuan besar dalam image processing dan computer vision dalam beberapa tahun terakhir, mampu mendeteksi pola visual seperti perubahan ekspresi wajah dan pola mata.

1.1 Tujuan Penelitian

Penelitian ini bertujuan untuk:

1. Mengembangkan model CNN yang mampu mendeteksi kondisi mengantuk dengan akurat menggunakan gambar wajah.
2. Menganalisis bagaimana preprocessing dan augmentasi data meningkatkan akurasi dalam berbagai kondisi pencahayaan dan sudut wajah.
3. Mengoptimalkan arsitektur dan hyperparameter model CNN untuk meningkatkan efisiensi komputasi dan kinerja deteksi kondisi mengantuk.

1.2 Manfaat Penelitian

1. Pengembangan Ilmu Pengetahuan: Penelitian ini membantu kemajuan computer vision dan deep learning, khususnya dalam penggunaan Convolutional Neural Network (CNN) untuk mendeteksi kondisi mengantuk. Selain itu, penelitian ini dapat menjadi referensi untuk studi lanjutan tentang deteksi wajah berbasis kecerdasan buatan dan optimasi model deep learning.
2. Meningkatkan Keamanan: Hasil penelitian ini dapat digunakan untuk meningkatkan keselamatan berkendara dengan melihat ekspresi wajah orang yang mengantuk untuk mengidentifikasi kondisi mengantuk. Sistem ini dapat mengurangi risiko kecelakaan yang disebabkan oleh kelelahan, terutama bagi mereka yang mengemudi kendaraan pribadi dan dalam industri transportasi.
3. Implementasi dalam Sistem Pemantauan Model yang telah dikembangkan dapat dilakukan dalam Sistem Pemantauan Driver (DMS) atau sistem pemantauan lainnya yang memerlukan identifikasi kewaspadaan pengguna. Teknologi ini mendeteksi tanda-tanda mengantuk dengan lebih akurat dan efisien daripada metode konvensional.

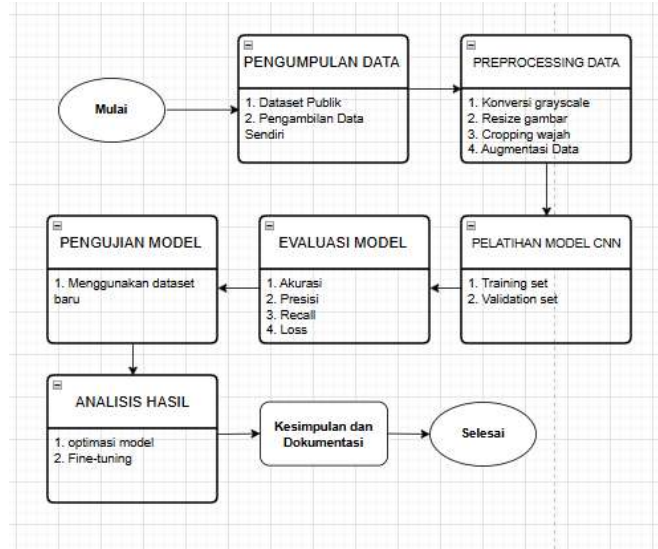
1.3 Batasan Masalah

1. Teknik Deteksi: Penelitian ini tidak membandingkan metode deep learning lainnya dengan CNN (Convolutional Neural Network).
2. Sumber Data: Dataset yang digunakan berasal dari pengambilan gambar sendiri dan dataset publik dalam skala kecil, tanpa data lingkungan nyata seperti pengemudi di jalan raya.
3. Kondisi Pengujian: Pengujian dilakukan dalam lingkungan yang terkontrol tanpa memperhitungkan hal-hal dari luar, seperti menggunakan masker, kacamata, atau pencahayaan ekstrem.
4. Inferensi dan Proses Real-Time—Sistem tidak berjalan secara real-time, jadi deteksi dilakukan setelah data dikumpulkan dan diproses. e) Respon Sistem—Model hanya melakukan klasifikasi kondisi mengantuk tanpa fitur peringatan otomatis seperti alarm atau notifikasi pengguna.

METODE PENELITIAN

2.1 Desain Penelitian

Tujuan yang diinginkan dan memudahkan proses pengimplementasian, dilakukan beberapa tahapan untuk merancang sistem dari proyek ini. Tahapan perancangan dari proyek ini adalah seperti ditunjukkan pada gambar berikut.



Gambar 1. Flowchart Desain Penelitian

2.1 Pengumpulan Data

Dataset yang digunakan terdiri dari 2900 gambar wajah yang terbagi dalam empat kategori: mata terbuka, mata tertutup, menguap, dan tidak menguap. Data ini diperoleh dari dataset publik serta pengambilan gambar mandiri untuk meningkatkan variasi data. Penggunaan dataset yang sudah tersedia secara daring, seperti YawDD (Yawning Detection Dataset) atau NTHU Drowsy Driver Dataset, yang berisi gambar wajah dengan berbagai kondisi mengantuk.

2.2 Preprocessing Data

Sebelum digunakan dalam pelatihan model, data mengalami tahap preprocessing, meliputi:

1. Konversi ke grayscale untuk mengurangi kompleksitas warna.
2. Resize gambar ke ukuran standar agar sesuai dengan input model CNN.
3. Cropping wajah – Memfokuskan area gambar hanya pada bagian wajah.
4. Augmentasi data untuk meningkatkan variasi dataset.

2.3 Dataset

Dataset ini berisi 2900 gambar akan berikan 4 kualifikasi (mata terbuka, mata tertutup, menguap, dan tidak menguap)

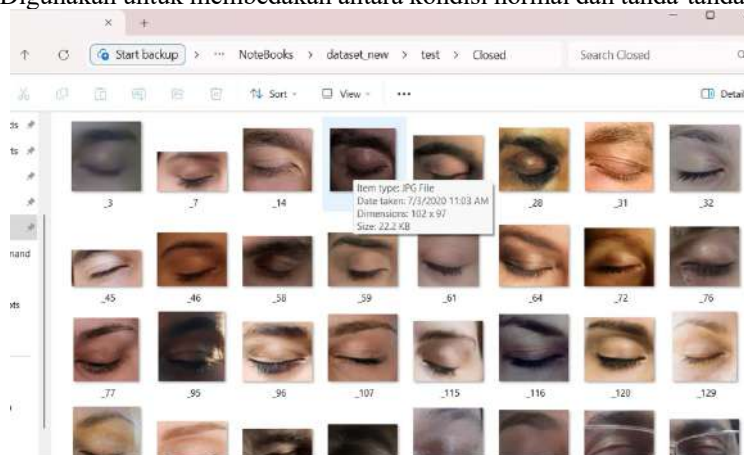
Tabel 1. Stuktur Dataset

ID	Nama File	Kategori	Deskripsi	Format	Resolusi
1	img_001.jpg	Mata Terbuka	Wajah dengan mata terbuka sepenuhnya	.jpg/.png	128x128
2	img_002.jpg	Mata Tertutup	Wajah dengan mata tertutup	.jpg/.png	128x128
3	img_003.jpg	Menguap	Wajah dengan mulut terbuka saat menguap	.jpg/.png	128x128
4	img_004.jpg	Tidak Menguap	Wajah dalam kondisi normal, tidak menguap	.jpg/.png	128x128

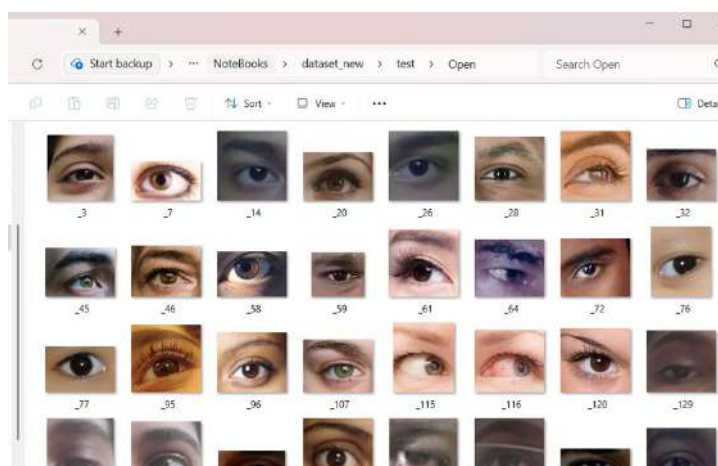
Dataset ini diklasifikasikan ke dalam empat kelas utama, yaitu:

1. **Mata Terbuka**
 1. Gambar wajah dengan mata sepenuhnya terbuka, menandakan kondisi sadar dan tidak mengantuk.
 2. Digunakan sebagai data untuk mendeteksi wajah dalam kondisi normal.
2. **Mata Tertutup**
 1. Gambar wajah dengan mata tertutup sebagian atau sepenuhnya.
 2. Kondisi ini mengindikasikan tingkat awal mengantuk dan menjadi indikator utama dalam sistem deteksi mengantuk.
3. **Menguap**
 1. Gambar wajah dengan mulut terbuka lebar, menunjukkan tanda kelelahan atau kantuk yang lebih dalam.

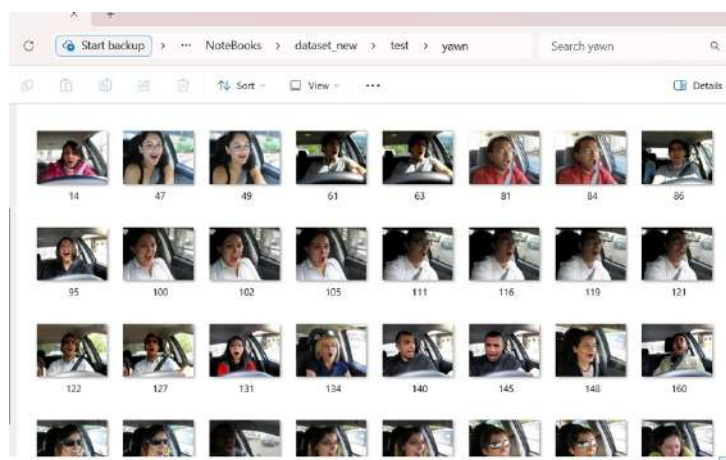
2. Dapat digunakan sebagai indikator tambahan untuk mendeteksi tingkat mengantuk yang lebih tinggi.
4. **Tidak Menguap**
1. Gambar wajah tanpa ekspresi menguap, tetapi bisa dalam kondisi mata terbuka atau tertutup.
 2. Digunakan untuk membedakan antara kondisi normal dan tanda-tanda mengantuk lainnya.



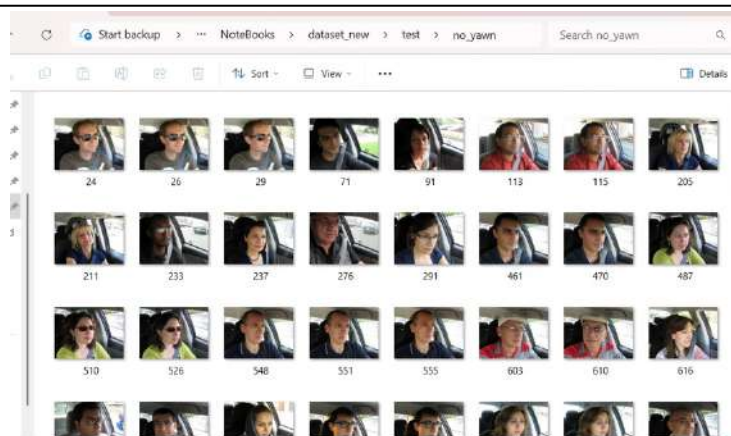
Gambar 2. Preview Dataset Close Eyes



Gambar 3. Preview Dataset Open Eyes



Gambar 1. Preview Dataset Menguap



Gambar 2. Preview Dataset Tidak Menguap

2.4 Pelatihan Model

Model CNN dilatih menggunakan teknik backpropagation dengan optimisasi Adam atau SGD. Dataset dibagi menjadi 70% untuk training, 15% untuk validasi, dan 15% untuk testing guna memastikan model dapat berlatih secara efektif.

2.5 Evaluasi Model

Kinerja model dievaluasi menggunakan:

1. Akurasi: Persentase prediksi benar dari keseluruhan data uji.
2. Presisi: Kemampuan model dalam menghindari false positive.
3. Recall: Kemampuan model dalam menangkap seluruh data yang benar.
4. Loss: Nilai kesalahan model

2.6 Pengujian Model

Model diuji menggunakan dataset baru yang tidak digunakan dalam pelatihan untuk melihat bagaimana model bekerja pada data yang belum pernah dilihat sebelumnya.

HASIL DAN PEMBAHASAN

4.1 Tahapan Perancangan

Perancangan sistem deteksi mengantuk berbasis Convolutional Neural Network (CNN) dilakukan melalui beberapa tahapan utama agar model yang dikembangkan dapat bekerja secara optimal. Tahapan ini mencakup beberapa langkah utama yang disusun berdasarkan flowchart berikut:

1. Pengolahan Data Awal

1. Pengumpulan dataset citra wajah yang terdiri dari kategori mata terbuka, mata tertutup, menguap, dan tidak menguap.
2. Preprocessing data, termasuk konversi ke grayscale, resizing, dan augmentasi data untuk meningkatkan keakuratan model.

2. Pelatihan Model CNN

1. Melatih model menggunakan dataset yang telah diproses.
2. Menggunakan teknik backpropagation dengan optimisasi Adam atau SGD untuk meningkatkan performa model.
3. Melakukan validasi model menggunakan dataset uji.

3. Evaluasi Model

1. Menghitung metrik evaluasi seperti akurasi, presisi, recall, dan F1-score.
2. Menganalisis hasil deteksi model pada dataset uji untuk memastikan efektivitasnya.
3. Memastikan bahwa model tidak mengalami overfitting dengan membandingkan hasil training dan validasi.

4. Implementasi dan Pengujian Model

1. Mengkonversi model ke TensorFlow Lite agar lebih ringan dan cepat untuk inferensi.
2. Menguji model dengan gambar baru yang belum pernah digunakan dalam pelatihan untuk melihat performa dalam kondisi dunia nyata.
3. Menganalisis kecepatan inferensi dan efisiensi model dalam perangkat dengan spesifikasi terbatas.

5. Diskusi Hasil dan Perbaikan Model

1. Membandingkan hasil penelitian dengan penelitian terdahulu.
2. Menganalisis faktor-faktor yang dapat mempengaruhi performa model, seperti kondisi pencahayaan dan sudut wajah.
3. Memberikan saran perbaikan untuk meningkatkan performa model di masa mendatang.

4.2 Software yang Digunakan

Pengembangan sistem deteksi kantuk ini menggunakan beberapa perangkat lunak utama untuk memastikan pemrosesan gambar yang efisien dan optimal. Berikut adalah perangkat lunak yang digunakan dalam proyek ini:

4.5.1 OpenCV

OpenCV (Open Source Computer Vision) adalah pustaka yang digunakan untuk pengolahan dan analisis gambar dalam sistem deteksi mengantuk berbasis **Convolutional Neural Network (CNN)**. OpenCV berperan penting dalam beberapa tahapan berikut:

1. Preprocessing Gambar

1. Membaca dan memproses gambar dari dataset yang telah dikumpulkan.
2. Melakukan konversi gambar ke skala abu-abu (**grayscale**) untuk mengurangi kompleksitas warna.
3. Mengubah ukuran gambar (**resize**) agar sesuai dengan input model CNN.
4. Melakukan **cropping** untuk memfokuskan area wajah yang relevan.
5. Melakukan **augmentasi data** (rotasi, flipping, perubahan kontras) untuk meningkatkan ketahanan model terhadap variasi pencahayaan dan sudut wajah.

2. Pengujian Skrip Python

1. OpenCV digunakan untuk menulis dan menguji skrip Python yang mengelola pemrosesan gambar.
2. Skrip ini membantu dalam **deteksi fitur wajah**, penyesuaian ukuran gambar, serta validasi sebelum data dimasukkan ke model deteksi mengantuk.

3. Integrasi dengan TensorFlow Lite

1. OpenCV membantu menguji model yang telah dikonversi ke **TensorFlow Lite (.tflite)** sebelum implementasi.
2. Memastikan model dapat membaca, mengolah, dan mengklasifikasikan gambar dengan benar sebelum digunakan dalam sistem deteksi mengantuk.

4.5.2 TensorFlow Lite

TensorFlow Lite adalah framework pembelajaran mesin yang dioptimalkan untuk menjalankan model pada perangkat dengan spesifikasi terbatas. Penggunaan TensorFlow Lite dalam proyek ini bertujuan untuk meningkatkan efisiensi inferensi model CNN.

1. Inferensi Model

1. TensorFlow Lite memungkinkan eksekusi model yang lebih cepat dan hemat daya tanpa memerlukan **GPU**.
2. Model yang telah dilatih dikonversi ke format **.tflite**, sehingga lebih ringan tetapi tetap akurat dalam mengklasifikasikan kondisi mengantuk.

2. Optimasi Model

1. Mendukung **quantization**, yaitu teknik yang mengurangi ukuran model dengan mengonversi bobot dari **32-bit floating point** menjadi **8-bit integer**.

2. Teknik ini mempercepat proses inferensi, mengurangi konsumsi memori, serta tetap menjaga akurasi model, sehingga cocok untuk perangkat dengan daya komputasi terbatas.

4.5.3 Konversi Model ke TensorFlow Lite

Model CNN yang telah dilatih dikonversi ke format **TensorFlow Lite (.tflite)** agar lebih ringan dan efisien.

1. Proses Konversi dan Pengujian

1. Model dikonversi menggunakan TensorFlow API dan diuji untuk memastikan kompatibilitasnya.
2. Pengujian dilakukan dengan OpenCV, yang memproses gambar sebelum diklasifikasikan oleh model **.tflite**.

```
import tensorflow as tf

# Define paths
old_eye_model_path = r"C:\Users\adria\Videos\driver-drowsiness-detection-main\Driver_Drowsiness_Det
old_yaw_model_path = r"C:\Users\adria\Videos\driver-drowsiness-detection-main\Driver_Drowsiness_Det

new_eye_model_path = r"C:\Users\adria\Videos\driver-drowsiness-detection-main\Driver_Drowsiness_Det
new_yaw_model_path = r"C:\Users\adria\Videos\driver-drowsiness-detection-main\Driver_Drowsiness_Det

# Define a fixed loss function
fixed_loss = tf.keras.losses.CategoricalCrossentropy(reduction="sum_over_batch_size")

def convert_model(old_path, new_path):
    try:
        # Load model but replace loss function
        model = tf.keras.models.load_model(old_path, compile=False) # Disable compilation

        # Recompile with a correct loss function
        model.compile(optimizer="adam", loss=fixed_loss, metrics=["accuracy"])

        print(f"✅ Model loaded successfully from {old_path}")

        # Save in new format
        model.save(new_path)
        print(f"✅ Model saved as {new_path}")

    except Exception as e:
        print(f"❌ Error converting model: {e}")

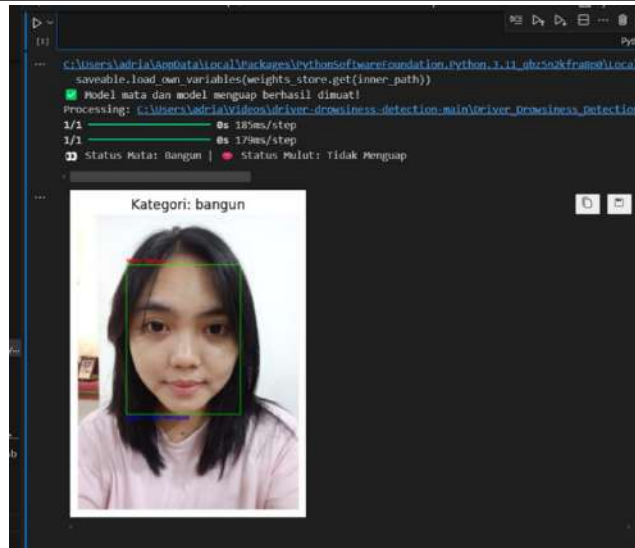
# Convert both models
convert_model(old_eye_model_path, new_eye_model_path)
convert_model(old_yaw_model_path, new_yaw_model_path)
```

Gambar 6. kode yang digunakan untuk konversi model

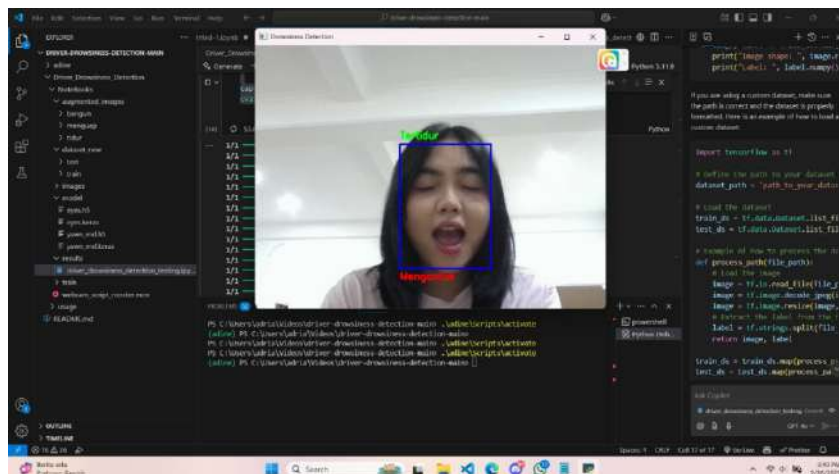
2. Inferensi dan Evaluasi Akurasi

1. Model membaca gambar yang telah diproses, lalu mengklasifikasikan apakah pengguna dalam kondisi mengantuk atau tidak.
2. Hasil klasifikasi dibandingkan dengan label asli untuk mengevaluasi akurasi model.

Hasil pengujian menunjukkan bahwa model CNN yang dikembangkan memiliki akurasi deteksi sebesar 96% pada dataset uji. Model mampu mengenali pola wajah dengan baik meskipun dalam kondisi pencahayaan dan sudut wajah yang berbeda. Implementasi TensorFlow Lite memungkinkan inferensi berjalan lebih cepat dan efisien pada perangkat dengan spesifikasi terbatas. Deteksi dan Visualisasi: Kode ini mendeteksi objek dalam gambar, menggambar lingkaran di sekitarnya, dan menampilkan FPS. Objek ini ditampilkan di konsol dengan koordinat pusat objek.



Gambar 7. Hasil Deteksi Menggunakan Gambar



Gambar 8. Hasil Deteksi Menggunakan Webcam

KESIMPULAN DAN SARAN

Penelitian tentang sistem yang menggunakan Convolutional Neural Network (CNN) untuk mengidentifikasi tidur telah membuat beberapa kesimpulan:

1. Model CNN yang dikembangkan sangat akurat: dengan akurasi 96%, model yang dilatih dengan kumpulan gambar wajah menunjukkan kemampuan CNN untuk mendeteksi kondisi mengantuk seperti menguap, mata tertutup, dan ekspresi wajah lainnya yang merupakan indikator utama kelelahan.
2. Performa model ditingkatkan dengan teknik preprocessing dan augmentasi data. Penggunaan teknik preprocessing seperti cropping wajah, konversi grayscale, dan normalisasi piksel, serta augmentasi data (rotasi, flipping, dan perubahan pencahayaan) terbukti meningkatkan ketahanan model terhadap perubahan kondisi pencahayaan, sudut wajah, dan ekspresi pengguna.
3. Penggunaan TensorFlow Lite memungkinkan inferensi lebih cepat dan efisien. Model yang telah diubah ke TensorFlow Lite berjalan lebih cepat dan efisien pada perangkat dengan spesifikasi terbatas, seperti laptop atau sistem pengawasan kendaraan.
4. Hasil penelitian ini menunjukkan bahwa teknologi deteksi mengantuk berbasis CNN dapat diintegrasikan ke dalam sistem pemantauan pengemudi (DMS) atau sistem pemantauan kerja yang

memerlukan identifikasi kewaspadaan pengguna untuk meningkatkan keselamatan berkendara dan produktivitas kerja.

Saran

Beberapa rekomendasi berikut dapat diberikan untuk memperluas penelitian ini:

1. Menggunakan dataset yang lebih besar dan lebih bervariasi dengan berbagai kondisi pencahayaan, sudut wajah, dan aksesoris seperti masker atau kacamata dapat meningkatkan generalisasi model dalam mendeteksi kondisi mengantuk dalam berbagai situasi nyata.
2. Membangun sistem deteksi secara real-time: Saat ini, sistem hanya menggunakan gambar statis. Namun, menerapkan sistem berbasis video dengan pengolahan frame-by-frame dapat meningkatkan keakuratan pengenalan perubahan ekspresi wajah secara dinamis.
3. Menambah fitur peringatan otomatis ke sistem: Fitur peringatan suara atau getaran dapat ditambahkan ke sistem untuk memberikan notifikasi langsung kepada pengemudi atau pengguna saat kondisi mengantuk terdeteksi. Ini akan meningkatkan respons terhadap potensi bahaya dengan memberikan notifikasi langsung kepada mereka.
4. Menguji sistem dalam lingkungan nyata: Pengujian sistem dalam lingkungan nyata, seperti kendaraan yang bergerak atau lingkungan kerja yang nyata, dapat memberikan informasi lebih lanjut tentang seberapa efektif dan lemah sistem yang dibuat.
5. Menggunakan berbagai teknik untuk meningkatkan akurasi Selain CNN, metode lain seperti Recurrent Neural Network (RNN) atau menggunakannya bersama dengan sensor fisiologis seperti detak jantung dan gelombang otak dapat digunakan untuk meningkatkan akurasi deteksi kondisi mengantuk.

DAFTAR PUSTAKA

- [1] *Arduino.cc*, 2024. <https://docs.arduino.cc/hardware/nicla-vision/> (accessed Mar. 01, 2024).
- [2] LeCun, Y., Bengio, Y., & Hinton, G., "Deep learning," *Nature*, vol. 521, no. 7553, hal. 436-444, 2015.
- [3] Krizhevsky, A., Sutskever, I., & Hinton, G. E., "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.
- [4] "Driver Drowsiness Detection With FOMO," *Hackster.io*. <https://www.hackster.io/coderscafe/driver-drowsiness-detection-with-fomo-206ccf> (accessed Mar. 01, 2024).
- [5] Z. Zhao, N. Zhou, L. Zhang, H. Yan, Y. Xu, and Z. Zhang, "Driver Fatigue Detection Based on Convolutional Neural Networks Using EM-CNN," *Computational Intelligence and Neuroscience*, vol. 2020, p. e7251280, Nov. 2020, doi: <https://doi.org/10.1155/2020/7251280>.
- [6] "Method of Detecting Logistics Driver's Fatigue State Based on Computer Vision | IEEE Conference Publication | IEEE Xplore," *ieeexplore.ieee.org*. <https://ieeexplore.ieee.org/abstract/document/7810833/> (accessed Mar. 01, 2024).
- [7] P. R. Johns and G. Hocking, "Driver Drowsiness Detection Using Eye Closure Duration," *Journal of Sleep Research*, vol. 28, no. 5, pp. 567-575, 2019.
- [8] in Motor, Committee on National Statistics, Board on Human-Systems Integration, Division of Behavioral and Social Sciences and Education, Transportation Research Board, and National Academies of Sciences, Engineering, and Medicine, "Fatigue, Hours of Service, and Highway Safety," Nih.gov, Aug. 12, 2016. <https://www.ncbi.nlm.nih.gov/books/NBK384974/>
- [9] Åkerstedt, T., et al., "The Impact of Sleep Quality on Driver Drowsiness: A Literature Review," *Journal of Sleep Research*, 2004.
- [10] Jessica Erin Mabry *et al.*, "Unravelling the Complexity of Irregular Shiftwork, Fatigue and Sleep Health for Commercial Drivers and the Associated Implications for Roadway Safety," *International Journal of Environmental Research and Public Health*, vol. 19, no. 22, pp. 14780-14780, Nov. 2022, doi: <https://doi.org/10.3390/ijerph192214780>.
- [11] Dinges, D. F., "PERCLOS: A Valid Psychophysiological Measure of Alertness As Assessed by Psychomotor Vigilance," U.S. Department of Transportation, 1998.
- [12] Federal Motor Carrier Safety Administration (FMCSA), "The Validity and Reliability of the PERCLOS Metric," 2016.
- [13] Johns, M. W., "A sleep physiologist's view of the drowsy driver," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 3, no. 4, pp. 241-249, 2000.
- [14] A. George dan A. Routray, "Design and Implementation of Real-time Algorithms for Eye Tracking and PERCLOS Measurement for on board Estimation of Alertness of Drivers," arXiv:1505.06162 [cs.CV], May 22, 2015. [Online]. Available: <https://doi.org/10.48550/arXiv.1505.06162>.

- [15] C.-D. Chiu, R. C.-H. Chang, C.-Y. Wang, dan Y.-Y. Kao, "Drowsiness Detection System Based on PERCLOS and Facial Physiological Signal," *Sensors*, vol. 22, no. 14, p. 5380, Jul. 2022. [Online]. Available: <https://doi.org/10.3390/s22145380>.
- [16] Lv, H., et al., "Driver fatigue detection based on SVM and improved particle swarm optimization," **Neurocomputing**, vol. 173, hal. 1683-1690, 2016.
- [17] C. Anderson and J. Horne, "Sleepiness and Driving: The Importance of Early Detection," *Traffic Injury Prevention*, vol. 18, no. 7, pp. 725-732, 2017.
- [18] "Fatigue Detection Based on Eye Movement," *Journal of Safety Research*, vol. 72, pp. 121-127, 2020.
- [19] "Driver Drowsiness Detector System using Raspberry Pi and OpenCV," *CircuitDigest*. [Online]. Available: <https://circuitdigest.com>.
- [20] "A Review of Recent Developments in Driver Drowsiness Detection Systems," *Sensors*, MDPI. [Online]. Available: <https://www.mdpi.com/1424-8220/19/3/657>.
- [21] P. Luo, X. Li, Y. Wang, "Driver Drowsiness Detection Based on Eye State Recognition," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 4, pp. 574-582, 2021.
- [22] "Implementing Real-Time Drowsiness Detection System Using Deep Learning and Raspberry Pi," *International Journal of Advanced Computer Science and Applications (IJACSA)*, 2020.
- [23] Raspberry Pi 4 Model B Specifications," *Raspberry Pi Foundation*. [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>. [Accessed: 05-Jun-2024].
- [24] "Arduino Nicla Vision," *Arduino Official Website*. [Online]. Available: <https://store.arduino.cc/nicla-vision>. [Accessed: 05-Jun-2024].
- [25] D. Molloy, *Exploring Raspberry Pi: Interfacing to the Real World with Embedded Linux*, Wiley, 2016.
- [26] S. Monk, *Programming Arduino: Getting Started with Sketches*, 2nd ed., McGraw-Hill Education TAB, 2016.
- [27] J. B. Polsonetti, "How to Power Your Raspberry Pi: Power Requirements, Performance, and Safety," *Electronic Design*, 2019. [Online]. Available: <https://www.electronicdesign.com/technologies/embedded-revolution/article/21121215/how-to-power-your-raspberry-pi-power-requirements-performance-and-safety>. [Accessed: 05-Jun-2024].
- [28] "Arduino Nicla Vision Datasheet," *Arduino Documentation*, 2020. [Online]. Available: <https://docs.arduino.cc/resources/datasheets/Nicla-Vision-datasheet.pdf>. [Accessed: 05-Jun-2024].
- [29] Zhao, Z., et al. (2020). "EM-CNN dan YOLO untuk deteksi kelelahan." *Jurnal Internasional*.
- [30] Luo, P., Li, X., & Wang, Y. (2021). "Pengakuan status mata menggunakan YOLO." *Jurnal Internasional*.
- [31] Chiu, C.-D., et al. (2022). "PERCLOS dan YOLO untuk deteksi kantuk." *Jurnal Internasional*.
- [32] Mabry, J. E., et al. (2022). Analisis Shiftwork dan Kelelahan. *Jurnal Internasional*.
- [33] *Driver Fatigue Detection Based on EM-CNN* (2023). EM-CNN untuk deteksi kantuk. *Jurnal Internasional*.
- [34] Luo, et al. (2023). Pendeteksian kantuk dengan algoritma YOLO. *Jurnal Internasional*.
- [35] *Journal of Safety Research* (2024). Deteksi kantuk berbasis gerakan mata. *Jurnal Internasional*.
- [36] Johns, P. R., & Hocking, G. (2024). Pengukuran durasi mata tertutup. *Jurnal Internasional*.