

Deteksi Kantuk Pengemudi Menggunakan Deep Learning

Salamat Nur Himawan¹, Robieth Sohiburroyan², Nur Budi Nugraha³

Jurusan Teknik Informatika, Politeknik Negeri Indramayu^{1, 2, 3}

snhimawan@polindra.ac.id¹, robieth.s@polindra.ac.id², nurbudinugraha@polindra.ac.id³

Abstract

One of the causes of driving accidents is driver fatigue. Driver fatigue can be measured by the level of sleepiness of the driver. In this study, a deep learning model was developed to detect sleepiness in drivers. The model will detect images of the eyelids and yawning state of the driver. The driver's level of sleepiness was grouped into not sleepy, sleepy and very sleepy. The developed model consists of a feature extractor in the form of several CNN layers, and a predictor in the form of FCN. CNN is able to extract features in the image so that it can provide better accuracy at the prediction stage. The model successfully learns from data and is able to detect driver sleepiness. The accuracy of the model reaches 0.932%. The failure of the model in detecting is caused by the wrong angle of taking the image to capture the front face. It is hoped that for the next research the number of datasets will be increased by all angles of image taking.

Keywords : Deep Learning, CNN, FCN, Kantuk

1. PENDAHULUAN

Kecelakaan lalu lintas merupakan suatu peristiwa yang tidak diduga dan tidak disengaja melibatkan kendaraan atau pengguna jalan. Kecelakaan terjadi sangat dipengaruhi oleh kondisi pengemudi kendaraan (IŞIK & ANAGÜN, 2021). Pada 2020, USA mencatat ada 633 total kecelakaan yang menyebabkan kematian diakibatkan oleh pengemudi yang mengantuk, total kecelakaan tersebut menyumbang 2.2% dari keseluruhan kecelakaan yang menyebabkan kematian (Stewart, 2022). Oleh karena itu, dibutuhkan pengembangan sistem keamanan untuk mencegah kecelakaan yang diakibatkan kantuk pengemudi.

Perkembangan teknologi dan ilmu pengetahuan yang semakin pesat memberikan kemudahan bagi kegiatan manusia (Nugraha & Alimudin, 2020). Salah satunya dengan membuat pendeteksian tingkat mengantuk pengemudi kendaraan bermotor (mobil). Beberapa metode telah dikembangkan untuk mendeteksi kantuk pengemudi dengan membuat penelitian mengukur pergerakan mata (Magán et al., 2022). Penelitian yang mengembangkan suatu metode yang dapat mendeteksi kantuk pada pengemudi dengan mengukur lamanya penutupan dan pembukaan kelopak mata (Robinson & Manikandan, 2018). Penelitian dengan melihat kondisi fisik pengemudi dengan mengukur denyut jantung, denyut nadi, gerakan kepala dan perilaku kendaraan menjadi faktor penentu dalam deteksi kantuk pengemudi (Padamata & Singothu, 2020).

Kedipatan mata seseorang berubah-ubah dalam waktu ke waktu, maka pendeteksi kantuk harus mengimplementasi metode yang dapat mengklasifikasi kedipan mata orang (Umut et al., 2017). Pada penelitian ini dikembangkan model *deep learning* untuk mendeteksi kantuk pada pengemudi. Deep learning tidak hanya digunakan untuk mengenali gambar lagi tetapi digunakan untuk mengatasi masalah lainnya, salah satunya adalah klasifikasi data temporal (Prasath et al., 2022). Dalam masalah pengenalan aktivitas manusia, aktivitas manusia bisa dikenali dari data sensor seperti barometer dan akselerometer yang datanya termasuk ke dalam data temporal. Model akan mendeteksi gambar kelopak mata dan kondisi menguap pada pengemudi.

2. TINJAUAN PUSTAKA

A. *Deep Learning*

Deep Learning merupakan salah satu bidang dari *machine learning* yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset yang besar. Teknik *Deep Learning* memberikan arsitektur yang sangat kuat untuk *supervised learning*. Dengan menambahkan lebih banyak lapisan maka model pembelajaran tersebut bisa mewakili data citra berlabel dengan lebih baik (Alzubaidi et al., 2021). Pada machine learning terdapat teknik untuk menggunakan ekstraksi fitur dari data pelatihan dan algoritma pembelajaran khusus untuk mengklasifikasi citra maupun untuk mengenali suara. Namun, metode ini masih memiliki beberapa kekurangan baik dalam hal kecepatan dan akurasi (Chirra et al., 2019).

Prinsip ini terus berkembang hingga *deep learning* semakin sering digunakan pada komunitas riset dan industri untuk membantu memecahkan banyak masalah data besar seperti *Computer Vision*, *Speech Recognition*, dan *Natural Language Processing*. *Deep learning* merupakan salah satu teknik dalam machine learning yang memiliki arsitektur yang lebih mendalam dibanding dengan teknik machine learning lainnya dalam menyelesaikan masalah prediksi maupun klasifikasi. Arsitektur umum *deep learning* adalah *Deep Neural Network* (DNN), *Deep Belief Network* (DBN), *Deep Convolutional Network* (DCN) (Patterson & Gibson, 2019).

B. *Convolutional Neural Network*

Dalam *Deep Learning*, *Convolutional Neural Network* (CNN) merupakan jenis algoritma yang biasa digunakan dalam pengenalan gambar dan klasifikasi gambar. CNN menggunakan masukan data berupa gambar (Kurniawan & Mustikasari, 2021). Komputer memproses masukan gambar berupa array yang merupakan representasi nilai pixel dari gambar. Dalam klasifikasi gambar, CNN akan memproses input gambar dan menggolongkannya ke dalam beberapa kategori, seperti mengkategorikan gambar tikus, kucing, anjing dan macan.

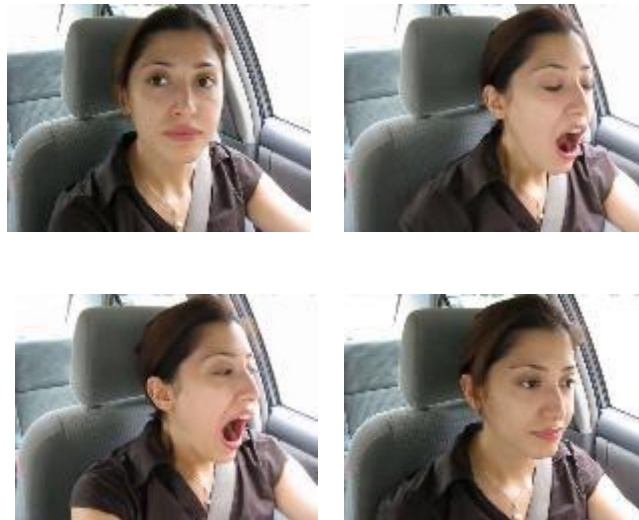
CNN merupakan pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi dalam bentuk citra. CNN ini termasuk kedalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra (Madhav et al., 2021). Pada dasarnya klasifikasi citra dapat digunakan dengan MLP, akan tetapi dengan metode MLP kurang sesuai untuk digunakan karena tidak menyimpan informasi spasial dari data citra dan menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik (Anton et al., 2021).

Secara teknis, CNN adalah sebuah arsitektur yang dapat dilatih dan terdiri dari beberapa tahap. Masukan (*input*) dan keluaran (*output*) dari setiap tahap adalah terdiri dari beberapa array yang biasa disebut *feature map*. Setiap tahap terdiri dari tiga layer yaitu konvolusi, fungsi aktivasi layer dan pooling layer (Rahim et al., 2020).

3. METODE PENELITIAN





A. Dataset

Pada penelitian ini data yang digunakan merupakan data citra wajah. Data citra diperbanyak dengan augmentasi citra. Augmentasi dapat berupa rotasi citra dan membalikan citra secara horizontal. Sebelum melakukan *training* pada Model, data citra dipotong dibagian wajah dan mata. Penelitian ini digunakan haarcascade hasil pengembangan OpenCV sebagai fitur ekstraksi untuk mendapatkan citra wajah dan mata.



Gambar 1. Data citra pengemudi

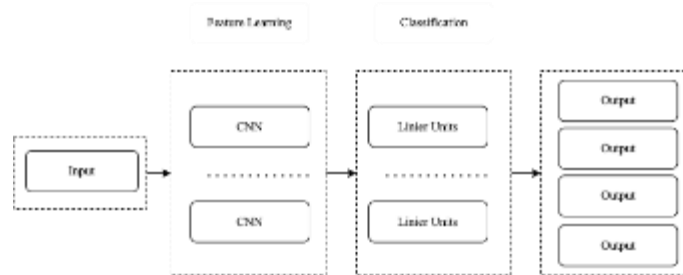
Tabel 1 Hasil augmentasi citra

Augmentasi	Sebelum	Sesudah
<i>Horizontal Flip</i>		
<i>Rotation</i>		

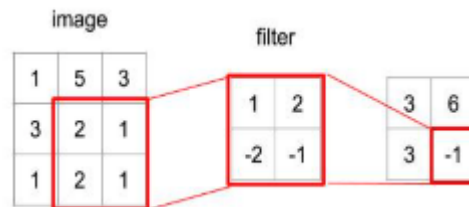
B. Arsitektur Model

Secara umum, hanya ada 2 bagian utama dalam arsitektur yang akan digunakan, yakni feature extractor berupa beberapa lapisan CNN, dan predictor berupa tumpukan FCN. Bagian pertama akan mengolah bundel data window yang masuk untuk membaca korelasi informasi antar tiap titik data dan menghasilkan suatu kumpulan fitur abstrak yang akan dimanfaatkan predictor untuk memprediksi output.

Pada tahap *feature extractor* menggunakan lapisan CNN. Citra dikonvolusi menggunakan sebuah filter dengan kernel yang bobotnya sudah ditentukan. Contoh konvolusi citra dapat dilihat seperti pada Gambar 2. Kemudian setelah itu proses tersebut menghasilkan output yang disebut *activation map* atau *feature map*.

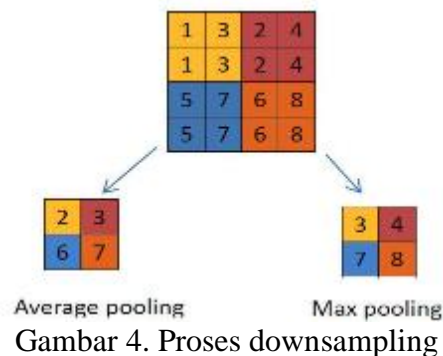


Gambar 2. Arsitektur model



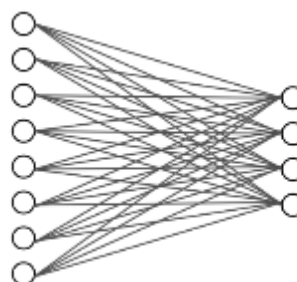
Gambar 3. Proses konvolusi

Setelah dilakukan konvolusi, citra sudah dipecah menjadi bagian yang kecil akan tetapi array yang dihasilkan masih terbilang cukup besar. Oleh karena itu, pada proses selanjutnya dilakukannya sebuah downsampling yang mana penggunaannya dinamakan *max pooling* atau *average pooling*. *Max pooling* mengambil nilai *pixel* paling besar di setiap pooling kernel dan *average pooling* mengambil nilai rata-rata *pixel* berdasarkan *pooling* kernel. Proses ini bertujuan untuk mengambil informasi terpenting dengan tidak mengurangi jumlah parameter. Berikut adalah ilustrasi pada downsampling.



Gambar 4. Proses downsampling

Pada proses terakhir yaitu *predictor*. *Predictor* terdiri dari lapisan *dense* yang merupakan *fully connected network*. Ilustrasi dari *fully connected network* dapat dilihat pada Gambar 5. Proses ini diawali dengan menginputkan array kecil yang sudah di downsampling ke dalam jaringan saraf lain. Hasil dari *predictor* merupakan klasifikasi dari mata terbuka atau tertutup dan menguap atau tidak menguap.



Gambar 5. Fully connected network

HASIL PENELITIAN DAN PEMBAHASAN

Setelah dataset berhasil terkumpul maka tahapan selanjutnya adalah melakukan *training* dan *testing*. Satu-satunya cara untuk mengetahui apakah model kita bagus atau tidak adalah dengan mengujinya pada kasus atau data baru yang belum dikenali oleh model. Pilihan yang lebih baik adalah dengan membagi dataset menjadi 2 bagian yaitu data *training* dan data *testing*.

Selanjutnya dilakukan pelatihan model pada *train set*, kemudian mengujinya pada *test set* sekumpulan data yang belum dikenali model. Membandingkan hasil prediksi dengan label sebenarnya dalam test set merupakan proses evaluasi performa model. Dengan menguji model terhadap data testing, kita dapat melihat kesalahan yang dibuat dan memperbaikinya. Data testing diambil dengan proporsi 70:30. Detail arsitektur model yang akan digunakan terlihat pada

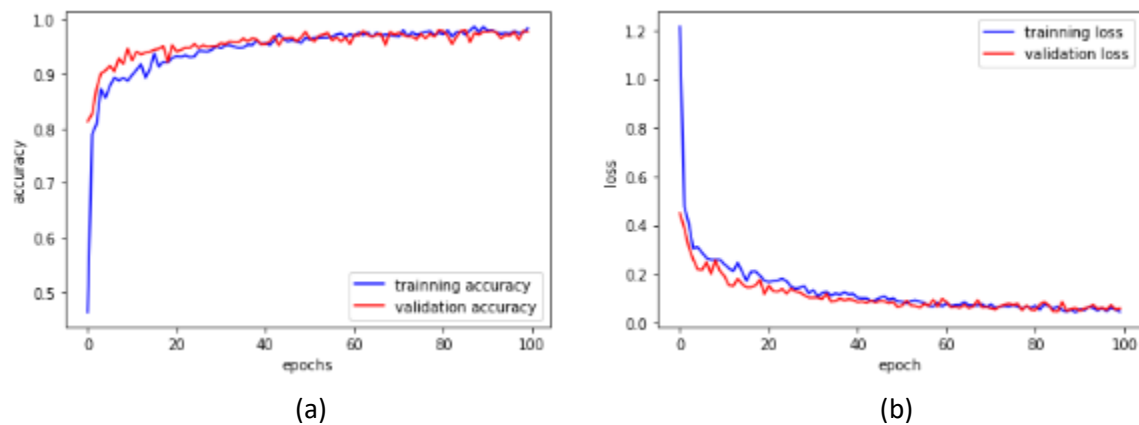
Tabel 2.

Tabel 2. Detail arsitektur model

Proses	Parameter
Convolution	Filter = 256, kernel = (3,3), Padding = 1, Shape (145,145,3), Activation = ReLU
Max Pooling 2D	Pool_size = (2,2)
Convolution	Filter = 128, kernel = (3,3), Padding = 1, Shape (71,71,256), Activation = ReLU
Max Pooling 2D	Pool_size = (2,2)
Convolution	Filter = 64, kernel = (3,3), Padding = 1, Shape (34,34,128), Activation = ReLU
Max Pooling 2D	Pool_size = (2,2)
Convolution	Filter = 32, kernel = (3,3), Padding = 1, Shape (16,16,64), Activation = ReLU
Max Pooling 2D	Pool_size = (2,2)
Flatten	
Dropout	0.5
Dense	Units = 64, Activation = ReLU
Dense	Units = 4, Activation = Softmax

Tabel 3. Data Hyperparameter

Hyperparameter	Value
Epoch	100
Optimizer	Adam
Initial learning rate	0.001
Decay rate	0.98



Gambar 6. (a) grafik akurasi berdasarkan epoch (b) grafik loss berdasarkan epoch

Pada penelitian ini, proses *training* model dilakukan dengan 100 epoch dengan inisial *learning rate* sebesar 0.001. *Learning rate* dibuat menurun mengikuti kenaikan epoch, *learning rate* menurun sebesar *decay rate*. *Hyperparameter* yang digunakan terlihat pada Tabel 3. Setelah dilakukan proses *training*, model di *test* menggunakan *test* data.

Berdasarkan Gambar 6a dan Gambar 6b dapat terlihat bahwa model dapat dengan baik mengklasifikasikan antara menguap, tidak menguap, mata tertutup dan mata terbuka. Nilai akurasi menaik dengan bertambahnya epoch, nilai akurasi mencapai 0.932% dan nilai *loss* menurun dengan bertambahnya epoch, nilai *loss* mencapai 0.121. Dari hasil *testing* dengan jumlah 680 citra didapatkan total 634 citra dapat diklasifikasi dengan benar. Hasil klasifikasi dapat dilihat pada tabel *confusion matrix* berikut.

Tabel 4. Confusion Matrix

Prediksi Aktual	Menguap	Tidak Menguap	Mata Tertutup	Mata Terbuka
Menguap	48	13	2	0
Tidak Menguap	0	70	4	0
Mata Tertutup	1	2	210	4
Mata Terbuka	0	0	20	306

Setelah melakukan klasifikasi menguap, tidak menguap, mata tertutup dan mata terbuka, model akan mendeteksi tingkat kantuk pengemudi. Tingkat kantuk pengemudi dibagi menjadi tiga bagian yaitu: tidak mengantuk, mengantuk dan sangat mengantuk, Pengemudi dikatakan tidak mengantuk ketika mata terbuka dan tidak menguap, pengemudi mengantuk ketika salah satu pengemudi menguap atau mata tertutup dan pengemudi sangat mengantuk ketika pengemudi menguap dan mata tertutup. Berikut merupakan contoh hasil deteksi kantuk pada pengemudi.

Terlihat pada contoh deteksi model dapat mendeteksi tingkat kantuk berdasarkan citra. Ada beberapa kegagalan dalam deteksi, terlihat pada Gambar 7 terlihat bahwa pengemudi sangat mengantuk tetapi model mendeteksi sebagai pengemudi yang tidak mengantuk. Kesalahan

model dalam mendeteksi dikarenakan sudut pengambilan citra yang tidak tepat menangkap muka bagian depan, sehingga model gagal mendeteksi citra tersebut.



Gambar 7. Hasil deteksi kantuk

KESIMPULAN DAN SARAN

Pada penelitian ini telah mempelajari kinerja CNN dalam mendeteksi kantuk pada pengemudi. CNN mampu mengekstraksi fitur-fitur pada citra sehingga dapat memberikan akurasi yang lebih baik pada tahap prediksi. Tingkat akurasi model mencapai 0.932%. Kegagalan model dalam mendeteksi dikarenakan sudut pengambilan citra yang tidak tepat menangkap muka bagian depan, sehingga model gagal mendeteksi citra tersebut. Saran untuk penelitian selanjutnya adalah diharapkan untuk penelitian berikutnya jumlah dataset diperbanyak dengan segala sudut pengambilan citra.

4. DAFTAR PUSTAKA

- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. In *Journal of Big Data* (Vol. 8, Issue 1). Springer International Publishing. <https://doi.org/10.1186/s40537-021-00444-8>
- Anton, A., Nissa, N. F., Janiati, A., Cahya, N., & Astuti, P. (2021). Application of Deep Learning Using Convolutional Neural Network (CNN) Method For Women's Skin Classification. *Scientific Journal of Informatics*, 8(1), 144–153. <https://doi.org/10.15294/sji.v8i1.26888>
- Chirra, V. R. R., Uyyala, S. R., & Kishore Kolli, V. K. (2019). Deep CNN: A machine learning approach for driver drowsiness detection based on eye state. *Revue d'Intelligence Artificielle*, 33(6), 461–466. <https://doi.org/10.18280/ria.330609>
- IŞIK, Ş., & ANAGÜN, Y. (2021). a Deep Learning Based Sleepness and Wakefulness Detection for Drivers. *Eskişehir Osmangazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi*, 29(3), 311–315. <https://doi.org/10.31796/ogummf.891255>
- Kurniawan, A. A., & Mustikasari, M. (2021). Implementasi Deep Learning Menggunakan

- Metode CNN dan LSTM untuk Menentukan Berita Palsu dalam Bahasa Indonesia. *Jurnal Informatika Universitas Pamulang*, 5(4), 544. <https://doi.org/10.32493/informatika.v5i4.6760>
- Madhav, T., Srivastava, A., & Kayalvizhi. (2021). A Deep Learning Approach To Detect Driver Drowsiness. *International Journal of Engineerig Research and Technology (IJERT)*, 10(05), 183–189.
- Magán, E., Sesmero, M. P., Alonso-Weber, J. M., & Sanchis, A. (2022). Driver Drowsiness Detection by Applying Deep Learning Techniques to Sequences of Images. *Applied Sciences (Switzerland)*, 12(3). <https://doi.org/10.3390/app12031145>
- Nugraha, N. B., & Alimudin, E. (2020). Mobile Application Development for Tourist Guide in Pekanbaru City. *Journal of Physics: Conference Series*, 1430(1). <https://doi.org/10.1088/1742-6596/1430/1/012038>
- Padamata, B. K., & Singothu, J. R. (2020). A Machine Learning Approach for Driver Drowsiness Detection. *International Journal of Engineering Research and Applications*, 10(11), 58–65. <https://doi.org/10.9790/9622-1011015865>
- Patterson, J., & Gibson, A. (2019). Deep Learning A Practioner's Approach. In *Journal of Chemical Information and Modeling* (Vol. 53, Issue 9).
- Prasath, N., Sreemathy, J., & Vigneshwaran, P. (2022). Driver Drowsiness Detection Using Machine Learning Algorithm. *8th International Conference on Advanced Computing and Communication Systems, ICACCS 2022, March*, 228–232. <https://doi.org/10.1109/ICACCS54159.2022.9785167>
- Rahim, A., Kusrini, K., & Luthfi, E. T. (2020). Convolutional Neural Network untuk Kalasifikasi Penggunaan Masker. *Inspiration: Jurnal Teknologi Informasi Dan Komunikasi*, 10(2), 109. <https://doi.org/10.35585/inspir.v10i2.2569>
- Robinson, L. T., & Manikandan, S. (2018). Drivers drowsiness measurement and the indication of eye movements through algorithmatic approach to avoid accidents. *Periodicals of Engineering and Natural Sciences*, 6(2), 210–223. <https://doi.org/10.21533/pen.v6i2.273>
- Umut, İ., Aki, O., Uçar, E., & Öztürk, L. (2017). Detection of Driver Sleepiness and Warning the Driver in Real-Time Using Image Processing and Machine Learning Techniques. *Advances in Science and Technology Research Journal*, 11(2), 95–102. <https://doi.org/10.12913/22998624/69149>