INFORMATIKA
Manajemen Informatika
Universitas Labuhanbatu
Vol. 10 No.1 / Januari/2022
2615-1855 (E-ISSN)
2303-2863 (P-ISSN)

Deteksi Kantuk Pengendara Mobil Berbasis Citra Menggunakan *Convolutional Neural Networks*

Habibullah Akbar
Prodi Magister Ilmu Komputer, Fakultas Ilmu Komputer
Universitas Esa Unggul
e-mail: habibullah.akbar@esaunggul.ac.id

Diah Aryani
Prodi Teknik Informatika, Fakultas Ilmu Komputer
Universitas Esa Unggul
e-mail: diah.aryani@esaunggul.ac.id

Suhandi Junaedi
Magister Teknik Informatika, *Binus Graduate Program*
Universitas Bina Nusantara
email: suhandi@binus.ac.id

ABSTRAK

Mengantuk bagi pengemudi dapat menyebabkan kecelakaan lalu lintas yang fatal. Banyak penelitian melaporkan bahwa gerakan yang berhubungan dengan mata dan menguap berkorelasi dengan risiko kelelahan dan keselamatan dalam berkendara. Namun, metode ini cenderung bergantung pada gerakan keadaan mata atau kondisi mulut. Dalam penelitian ini, kami menyajikan pendekatan berbasis *Convolutional Neural Network* untuk mendeteksi kantuk pengemudi secara otomatis tanpa perlu memodelkan kondisi lingkungan ataupun fitur wajah pengendara. Dataset citra yang digunakan diturunkan dari dataset video YawDD dimana resolusi yang digunakan adalah 32 x 32 piksel. Metode CNN yang digunakan adalah AlexNet yang memiliki dua lapisan konvolusi dan dibandingkan dengan metode tradisional yang masih harus melakukan pemilihan dan ekstrasi fitur secara manual. Eksperimen menunjukkan parameter terbaik yaitu *minibatch* senilai 30, *learning rate* senilai 0,1, rasio *training* dan *testing* yaitu 0,9 : 0,1, *dropout* senilai 10%, dan *epoch* senilai 500. Akurasi yang dihasilkan berhasil mencapai 77,8% walaupun waktu *training* yang dibutuhkan masih relatif tinggi. Meskipun demikian, metode yang ini mampu mengungguli metode tradisional yang masih memerlukan pemodelan fitur secara eksplisit (yaitu PERCLOS).

Kata kunci : Convolutional Neural Network, AlexNet, PERCLOS, Deteksi Kantuk, Pengendara Mobil.

ABSTRACT

Drowsiness for drivers can lead to fatal traffic accidents. Many studies report that eye movement and yawning are correlated with fatigue risk and driving safety. However, this method tends to depend on the state of the eye movement or the condition of the mouth. In this study, we present a Convolutional Neural Network-based approach to detect driver drowsiness automatically without the need to model environmental conditions or driver

INFORMATIKA
Manajemen Informatika
Universitas Labuhanbatu
Vol. 10 No.1 / Januari/2022
2615-1855 (E-ISSN)
2303-2863 (P-ISSN)

facial features. The image dataset used is derived from the YawDD video dataset where the resolution used is 32 x 32 pixels. The CNN method used is AlexNet which has two convolution layers and compared to the traditional method that require selection and extraction of features manually. The experiment showed the best parameters, namely minibatch 30, learning rate 0.1, training and testing ratio 0.9: 0.1, dropout 10%, and epoch 500 was able to produce accuracy upto 77.8% even though the training time required was still relatively high. However, this method was able to outperform traditional methods which still require explicit feature modeling (i.e. PERCLOS).

Keywords: Convolutional Neural Network, AlexNet, PERCLOS, Drowsiness Detection, Car Driver.

1. PENDAHULUAN

Kelelahan pengemudi selama mengemudikan mobil telah dianggap sebagai masalah utama karena meningkatnya jumlah kecelakaan di jalan. Dengan jadwal kehidupan yang padat saat ini, kelelahan saat mengemudi bisa menjadi lebih buruk terutama untuk kota-kota yang sibuk. Di Indonesia saja, tiga orang meninggal di jalan setiap jamnya (Satria *et al.*, 2020). Untuk mengatasi masalah ini, Pemerintah menargetkan penambahan jalan baru sepanjang 3.650 km termasuk 1.000 km jalan tol (PWC, 2016). Pemerintah Indonesia berencana mengalokasikan US$ 67,9 miliar untuk pembangunan infrastruktur jalan pada 2019 yang berlanjut menjadi US$ 429 miliar pada 2024. Pembangunan infrastruktur jalan dapat mengurangi kemacetan lalu lintas. Namun, selain karakteristik jalan tersebut, perilaku pengemudi (mengebut berlebihan, mabuk, dan melanggar lampu lalu lintas) juga bertanggung jawab atas kecelakaan kendaraan. Hal ini berimplikasi bahwa kita masih membutuhkan mekanisme kontrol bagi pengemudi untuk mengurangi kecelakaan lalu lintas.

Menurut (Williamson *et al.*, 2011; Soares *et al.*, 2020), terdapat hubungan yang kuat antara kelelahan dan risiko keselamatan dalam berkendara. Kelelahan dapat dipengaruhi oleh kesehatan dan masalah yang berhubungan dengan tidur. Kelelahan yang ekstrim dapat menyebabkan pengemudi mengantuk yang telah dianggap sebagai penyebab kecelakaan di jalan dan dapat menyebabkan cedera parah, dan risiko kematian yang tinggi. Dalam hal ini, kantuk mengacu pada penurunan atau hilangnya kewaspadaan yang menyebabkan pengemudi tertidur saat mengemudi.

Perkembangan deep learning yaitu jaringan saraf tiruan yang dapat melakukan ekstraksi fitur langsung dari data mentah telah menarik minat penelitian di bidang berbagai bidang. Berbagai macam convolutional neural networks (CNN) telah dikembangkan untuk klasifikasi adalah GoogLeNet dan AlexNet telah diterapkan untuk klasifikasi gambar (Żejmo Michałand Kowal et al., 2017), (Ciresan et al., 2013).

Pada penelitian ini, kami akan membahas penggunaan CNN untuk deteksi kantuk ataupun kelelahan saat mengemudi. Hal ini penting sebab kantuk itu sendiri berkaitan dengan bahaya semisal kecelakaan lalu lintas, keselamatan kerja, ataupun masalah lain yang memerlukan kondisi tubuh yang prima.

(Viola & Jones, 2001)Untuk mendeteksi kantuk, terdapat beberapa kelas metode yang dapat digunakan baik dengan sensor fisiologis, sensor kemudi, sensor kecepatan kendaraan dan kamera. Diantara sensor-sensor tersebut, penggunaan kamera

INFORMATIKA
Manajemen Informatika
Universitas Labuhanbatu
Vol. 10 No.1 / Januari/2022
2615-1855 (E-ISSN)
2303-2863 (P-ISSN)

yang paling mudah dilakukan untuk instalasi sistem deteksi kantuk. Citra yang dihasilkan dari kamera kemudian dapat menjadi data untuk menentukan apakah pengendara mengantuk atau tidak. Banyak peneliti yang menggunakan algoritma Viola-Jones (Viola and Jones, 2001) untuk mendeteksi wajah terlebih dahulu, kemudian melanjutkan untuk mengidentifikasi posisi mata, mulut maupun fitur yang lain pada wajah.

1.1 Penelitian Terkait

Permasalahannya, dari penelitian-penelitian yang telah dilakukan sebelumnya, banyak metode deteksi kantuk dilakukan berdasarkan ekstraksi fitur yang ditentukan secara manual seperti fitur mata seperti PERCLOS (Trutschel *et al.*, 2011), (Junaedi and Akbar, 2018). Selain mata dan mulut, fitur wajah lain seperti alis dan raut wajah juga dapat digunakan untuk mendeteksi kantuk pada pengendara kendaraan. Namun dengan kondisi background pengemudi yang sangat variatif maka fitur yang dibutuhkan juga akan berbeda-beda.

Saat ini, metode ekstraksi fitur manual sudah mulai ditinggalkan dan digantikan dengan metode *Convolutional Neural Network* (CNN) yang otomatis. Artinya fitur dapat dilatih secara otomatis didalam lapisan konvolusi *deep learning*. Pada penelitian (Jabbar *et al.*, 2020) telah mengembangkan CNN berbasis *Facial Landmark Detection* yang dirancang untuk perangkat Android. Akurasi yang dihasilkan dapat mencapai 83%. Namun model ini akan mengalami kesulitan jika pencahayaan lingkungan kurang baik dimana fitur wajah sulit didapatkan.

Selain itu, (Dua *et al.*, 2021) mengembangkan sebuah arsitektur CNN yang cukup kompleks. Arsitektur tersebut terdiri dari AlexNet, VGG-FaceNet, FlowImageNet dan ResNet. Akurasi yang dihasilkan berhasil mencapai 85%. Namun demikian, ke-empat CNN tersebut memiliki peran yang berbeda-beda. Model AlexNet digunakan untuk memodelkan latar belakang dan perubahan lingkungan seperti *indoor*, *outdoor*, siang dan malam hari. VGG-FaceNet digunakan untuk mengekstrak karakteristik wajah seperti etnisitas gender. FlowImageNet digunakan untuk fitur perilaku dan gerakan kepala, sedangkan ResNet digunakan untuk memodelkan gerakan tangan.

Walaupun berhasil mencapai akurasi melebihi 80%, kedua penelitian tersebut masih secara implisit memodelkan keadaan pencahayaan lingkungan. Oleh karena itu, timbul pertanyaan bagaimana performa CNN sekiranya digunakan pada data mentah citra ataupun video tanpa memodelkan berbagai karakteristik seperti lingkungan, kondisi wajah, gerakan tangan, dan sebagainya.

Pada penelitian ini kami tertarik untuk melihat bagaimana performa CNN untuk citra mentah tanpa perlu memodelkan karakteristik-karakteristik tersebut secara khusus. Artinya bagaimana caranya mendeteksi pengemudi yang sedang mengantuk menggunakan fitur yang dapat ditemukan secara otomatis menggunakan CNN.

2. METODE PENELITIAN

2.1 Pengumpulan Data

Dataset yang digunakan berasal dari dataset publik YawDD (Abtahi *et al.*, 2014), yang secara khusus berisi para pengemudi mobil yang menguap. Ada yang menggunakan kacamata maupun tidak, pengemudi sedang menguap, sedang berbicara dan normal mengemudikan

INFORMATIKA
Manajemen Informatika
Universitas Labuhanbatu
Vol. 10 No.1 / Januari/2022
2615-1855 (E-ISSN)
2303-2863 (P-ISSN)

kendaraan. Variasi latar belakang atau pencahayaan antara lain, mengemudi dalam kondisi terang, dan agak gelap. Pada penelitian ini, kami memfokuskan melakukan *training* CNN hanya pada dataset pria saja. Dari video, kami mengekstrak gambar para pengemudi dengan jumlah berikut ini:

- Citra Pengendara Mengantuk (total sampel 58)
- Citra Pengendara Tidak Mengantuk (total sampel 118)

Contoh sampel pengendara yang diturunkan dari dataset YawDD dapat dilihat pada Gambar 1.

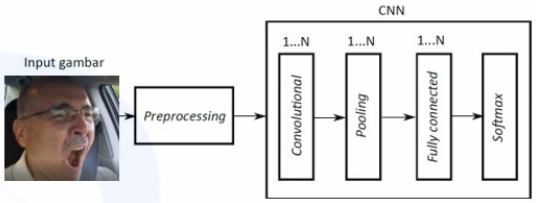
Gambar 1. Beberapa contoh sampel pengendara mobile pada YawDD dataset (Abtahi *et al.*, 2014).

2.2 Pra-pemrosesan Citra

Sebelum diproses, dataset yang diturunkan dari YawDD diubah resolusinya menjadi 32 x 32 piksel. Pemilihan nilai resolusi ini sesuai dengan penelitian sebelumnya dimana peningkatan ukuran resolusi tidak berpengaruh pada akurasi namun justru mengakibatkan meningkatnya waktu komputasi (Akbar, Anwar, et al., 2021).

2.3 Deteksi Kantuk menggunakan CNN

Dataset yang digunakan berasal dari dataset publik YawDD (Abtahi *et al.*, 2014), yang secara khusus berisi para pengemudi mobil yang menguap, ada yang menggunakan kacamata, dan ada yang berbicara. Metode CNN yang digunakan pada penelitian ini dapat dilihat pada gambar 2. Model ini berasal dari AlexNet yang disusun oleh (Krizhevsky, Sutskever and Hinton, 2012).



Gambar 2. Struktur umum lapisan CNN model AlexNet.

Model ini terdiri dari beberapa lapisan berikut yang mencakup:

1. *Input patch*. Data masukan yang dapat diterima CNN adalah input citra atau patch citra yang berasal dari dataset.
2. *Blocks*. Setiap blok CNN terdiri atas 3 lapisan yaitu lapisan konvolusi, ReLU, dan *pooling*.
   a. Lapisan konvolusi menerima input patch yang terdiri dari piksel-piksel. Lapisan ini terdiri dari kumpulan filter yang diinisialisasi secara acak untuk mencari representasi fitur dari suatu gambar berdasarkan kategori label penyakit. Setiap filter mengandung matrik saraf sebagai receptive field yang dimana nilai setiap sel saraf akan dilatih untuk mendeteksi fitur dari yang paling sederhana seperti edge, curve hingga parts.
   b. Lapisan ReLU akan menentukan apakah sinyal dari lapisan konvolusi

INFORMATIKA
Manajemen Informatika
Universitas Labuhanbatu
Vol. 10 No.1 / Januari/2022
2615-1855 (E-ISSN)
2303-2863 (P-ISSN)

dapat diteruskan pada lapisan berikutnya atau tidak. Bentuk fungsi ReLU dapat dinyatakan dengan *f(x)=max(0,x)* yang artinya akan memotong sinyal input yang memiliki nilai kurang dari 0.

c. Lapisan *pooling* akan mengurangi beban komputasi dengan cara menurunkan ukuran gambar yang diteruskan dari lapisan ReLU dengan melakukan down sampling.

3. Lapisan *fully-connected* merupakan lapisan perceptron yang menghimpun seluruh sinyal lapisan sebelumnya lalu memproses jumlah sinyal tersebut menggunakan fungsi softmax. Jumlah perceptron yang digunakan biasanya 4096 dan akan diturunkan sesuai dengan jumlah label yang ada. Label berisi kategori kantuk dan tidak yang diberikan pada dataset.

Dasar model AlexNet yang dikembangkan adalah convolutional neural networks yang telah tersusun dari lapisan dua lapis konvolusi, ReLU, *pooling*, fully connected, dan softmax yaitu model AlexNet2. Detil arsitektur model ini dapat dilihat pada (Akbar, Anwar, et al., 2021).

2.4 Evaluasi Performa CNN

Performa CNN dievaluasi berdasarkan akurasi yang diperoleh dari matriks *confusion* sebagaimana yang ditampilkan pada table 1.

Tabel 1. Matriks *Confusion*

Mengantuk	Prediksi	
	ya	tidak
ya	x_{11}	x_{12}
tidak	x_{21}	x_{22}

Nilai akurasi dapat diperoleh dengan membandingkan jumlah hasil prediksi yang benar terhadap ketiga jenis tersebut. Total positif sejati dapat dihitung dengan menggunakan (1).

$$TTP_{all} = \sum_{j=1}^{2} x_{jj} \quad (1)$$

Variabel x_{11} adalah total *true positif* yang artinya hasil prediksi pengendara adalah mengantuk sesuai dengan kondisi aslinya; sedangkan x_{22} adalah *true negatif* yaitu hasil prediksi pengendara tidak mengantuk sesuai kondisi aslinya. Adapun x_{12} adalah *false negatif* dimana hasil prediksi salah dengan mengklasifikasi data yang seharusnya mengantuk sebagai data yang mengantuk. Sebaliknya untuk x_{21} yaitu hasil prediksi adalah mengantuk padahal data labelnya tidak mengantuk. Akurasi dapat dihitung dengan menggunakan (2).

$$A = \frac{TTP_{all}}{All} \quad (2)$$

Kombinasi parameter awal CNN yang digunakan didasarkan pada waktu komputasi yang relatif singkat dan penelitian sebelumnya yaitu (Akbar, Anwar, et al., 2021):

- *epoch* = 10,
- *learning rate* = 0,0001,
- resolusi citra = 32 x32 piksel,
- rasio training-testing = 0,9: 0,1,
- arsitektur CNN = AlexNet2

Selain itu juga, performa CNN juga akan dibandingkan dengan metode tradisional yang masih menggunakan ekstraksi fitur PERCLOS secara manual. Detil metode ini dapat dilihat di (Junaedi and Akbar, 2018). Perhitungan PERCLOS adalah:

$$PERCLOS = \frac{N_t - N_s}{N_t} \times 100\% \quad (3)$$

INFORMATIKA
Manajemen Informatika
Universitas Labuhanbatu
Vol. 10 No.1 / Januari/2022
2615-1855 (E-ISSN)
2303-2863 (P-ISSN)

3. HASIL DAN BAHASAN

Eksperimen dilakukan pada perangkat komputer *Intel Core* i5-6500 CPU@3.2GHz dengan RAM sebesar 20 GB. Software yang digunakan adalah Matlab dengan *toolbox deep learning* yang berada diatas Windows 10 64-bit.

Tabel 2 menunjukkan peforma AlexNet berdasarkan parameter *minibatch*. Pada *minibatch* 10 dan 20 akurasi pada AlexNet hanya 61.1%. Pada *minibatch* 30, akurasi AlexNet meningkat menjadi 72.2% dengan waktu training 38.3 detik dan pada *minibatch* yang lebih tinggi akurasi justru semakin menurun. Nilai *minibatch* terbaik yaitu 30 ini kemudian digunakan untuk eksperimen berikutnya.

Tabel 2. Pencarian *minibatch* terbaik

No	*Minibatch*	Akurasi (%)	CPU (detik)
1	10	61,1	44,1
2	20	61,1	37,6
3	**30**	**72,2**	**38,3**
4	40	55,6	33,0
5	50	61,1	38,3
6	100	66,7	28,7

Dalam pengujian parameter *learning rate* sebagaimana yang dapat dilihat pada Tabel 3, kami menggunakan nilai dari 0.00001 hingga 1 yang dinaikkan dengan kelipatan 10. Secara teoritis, semakin tinggi nilai *learning rate* berarti semakin besar fluktuasi akurasi dari proses *training*. Berdasarkan eksperimen, nilai akurasi mencapai 66,7%, kecuali untuk nilai *learning rate* 0,001 yang hanya mencapai 61,1%. Namun demikian, waktu CPU terbaik adalah 38,1 detik yang dihasilkan oleh nilai *learning rate* 0,1. Nilai ini digunakan untuk eksperimen berikutnya.

Tabel 3. Pencarian *learning rate* terbaik (minibatch = 30)

Tabel 4 menunjukkan performa AlexNet berdasarkan rasio data *training* dan *testing* yang digunakan. Kami mengatur berbagai data *testing* dari 50% hingga 90%, dengan kenaikan level 10%. Akurasi terbaik mencapai 72,2% yang dihasilkan oleh rasio data *training* dan *testing* sebesar 0,9: 0,1 dengan waktu training 38,6 detik. Hasil eksperimen menampakkan bahwasanya semakin kecil data *training*, nilai akurasi juga semakin menurun.

No	*learning rate*	Akurasi (%)	CPU (detik)
1	0,00001	66,7	38.3
2	0,0001	66,7	38,3
3	0,001	61,1	38,3
4	0,01	66,7	38,3
5	**0,1**	**66,7**	**38,1**
6	1	66,7	38,4

Tabel 4. Pencarian rasio data *training* dan data *testing* (*learning rate* = 0.1)

No	*Training-testing*	Akurasi (%)	CPU time (seconds)
1	0,5 : 0,5	50,0	35,5
2	0,6 : 0,4	65,7	38,1
3	0,7 : 0,3	67,3	39,2
4	0,8 : 0,2	66,7	36,4
5	0,9 : 0,1	72,2	38,6

Tabel 5 menunjukkan performa AlexNet berdasarkan parameter *dropout* yang berfungsi sebagai regulator agar tidak terjadi *overfitting* pada proses *training*. Nilai dropout yang diuji dimulai dari 10 hingga 70 dengan kenaikkan 10. Hasil terbaik diberikan oleh nilai *dropout* 10% dengan waktu *training* 38,3 detik. Adapun untuk nilai *dropout* lainnya, akurasi tidak dapat bertumbuh dari 66,7%.

INFORMATIKA
Manajemen Informatika
Universitas Labuhanbatu
Vol. 10 No.1 / Januari/2022
2615-1855 (E-ISSN)
2303-2863 (P-ISSN)

Tabel 5. Pencarian parameter dropout (rasio data training dan testing = 0.9:0.1)

No	*Dropout* (%)	Akurasi (%)	CPU (detik)
1	**10**	**72,2**	**38,3**
2	20	66,7	38,5
3	30	66,7	38,2
4	40	66,7	38,3
5	50	66,7	38,4
6	60	66,7	38,8
7	70	66,7	39,0

Tabel 6 menunjukkan performa AlexNet berdasarkan parameter *epoch* atau iterasi. Kami menggunakan nilai dari 20 hingga 500. Hingga penggunaan *epoch* 300, akurasi tidak dapat melebihi 72,2%. Ketika nilai epoch dinaikkan hingga 500 iterasi, akurasi mampu mencapai 77,8%. Namun demikian waktu *training* yang dibutuhkan mencapai 1582,6 detik.

Tabel 6. Pencarian parameter *epoch* (*dropout* = 10%)

No	epoch	Akurasi (%)	CPU (detik)
1	20	55,6	69,5
2	30	55,6	100,8
3	40	72,2	131,4
4	50	61,1	164,4
5	100	55,6	323,2
6	300	55,6	954,9
7	**500**	**77,8**	**1582,6**

Tabel 7 menunjukkan perbandingan performa 3 model arsitektur AlexNet dengan metode tradisional yang menggunakan ekstraksi fitur manual PERCLOS (Junaedi and Akbar, 2018). Hasil akurasi terbaik diberikan oleh arsitektur 2BlokAlexNet dengan parameter optimal berikut:

- resolusi citra input = 32 x32 piksel,
- *minibatch* = 30,
- *learning rate* = 0,1,
- rasio *training*:*testing* = 0,9 : 0,1
- *dropout* = 10%,
- *epoch* = 500,

Kombinasi parameter tersebut mampu mengungguli akurasi metode tradisional yang masih menggunakan ekstraksi fitur manual PERCLOS sebesar 2,1%.

Tabel 7. Perbandingan metode deteksi kantuk usulan dengan penelitian sebelumnya.

[a] epoch = 500

No	Model	Akurasi (%)	CPU (detik)
1	AlexNet[a]	61,1	1530,0
2	**AlexNet2[a]**	**77,8**	**1582,6**
3	AlexNet3[a]	72,2	1641,3
4	PERCLOS (P60)[b]	75,7	-
5	PERCLOS (P70)[b]	65,3	-
6	PERCLOS (P80)[b]	53,8	-

[b] (Junaedi and Akbar, 2018)

Meskipun model AlexNet2 telah berhasil menggungguli metode tradisional yang masih menggunakan ekstraksi fitur tradisional seperti PERCLOS, namun model ini masih dibawah *framework* CNN yang lebih kompleks seperti pada (Jabbar *et al.*, 2020; Dua *et al.*, 2021).

4. KESIMPULAN DAN SARAN

Penelitian ini telah mempresentasikan metode deteksi kantuk berdasarkan citra menggunakan metode ekstraksi fitur otomatis menggunakan CNN yaitu model AlexNet. Model AlexNet ini memiliki 2 lapisan konvolusi yang nampaknya

INFORMATIKA
Manajemen Informatika
Universitas Labuhanbatu
Vol. 10 No.1 / Januari/2022
2615-1855 (E-ISSN)
2303-2863 (P-ISSN)

membuat model ini mampu mencapai akurasi 77,8% yang secara langsung memproses citra mentah. Keuntungan model ini adalah kemampuan mencapai akurasi tersebut tanpa harus memodelkan karakteristik lingkungan maupun objek pengendara. Namun demikian, model ini membutuhkan proses *training* yang cukup lama yaitu 1582,6 detik. Untuk memperbaiki hal ini, kedepannya kami menyarankan penggunaan hardware yang lebih baik ataupun dapat menggunakan metode optimisasi untuk mempercepat proses *training*.

DAFTAR PUSTAKA

Abtahi, S. et al., 2014 "YawDD: A yawning detection dataset," in *Proceedings of the 5th ACM multimedia systems conference*, pp. 24–28.

Akbar, H. et al. 2021 "Optimizing AlexNet using Swarm Intelligence for Cervical Cancer Classification," in 2021 Int*ernational Symposium on Electronics and Smart Devices* (ISESD), pp. 1–6.

Dua, M. et al. 2021 "Deep CNN models-based ensemble approach to driver drowsiness detection," *Neural Computing and Applications*, 33(8), pp. 3155–3168.

Jabbar, R. et al. 2020 "Driver drowsiness detection model using convolutional neural networks techniques for android application," in 2020 IEEE *International Conference on Informatics, IoT, and Enabling Technologies* (ICIoT), pp. 237–242.

Junaedi, S. and Akbar, H. 2018 "Driver drowsiness detection based on face feature and PERCLOS," *in Journal of Physics: Conference Series*, p. 12037.

Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012 "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, 25, pp. 1097–1105.

PWC, 2016 "Indonesian Infrastructure Stable foundations for growth," https://www.pwc.com/id/en/cpi/asset/indonesian-infrastructure-stable-foundations-for-growth.pdf.

Satria, R. et al., 2020 "A combined approach to address road traffic crashes beyond cities: hot zone identification and countermeasures in Indonesia," *Sustainability*, 12(5), p. 1801.

Soares, S. et al., 2020 "Analyzing driver drowsiness: From causes to effects," *Sustainability*, 12(5), p. 1971.

Trutschel, U. et al. 2011 "PERCLOS: An alertness measure of the past."

Viola, P. and Jones, M. 2001 "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition*. CVPR 2001, pp. I–I.

Williamson, A. et al. 2011 "The link between fatigue and safety," *Accident Analysis & Prevention*, 43(2), pp. 498–51.