sudo
Jurnal Teknik Informatika
https://jurnal.ilmubersama.com/index.php/sudo

Deep Learning

Deteksi Kantuk Pengemudi Berdasarkan Keterbukaan Mata Menggunakan Model Ringan dari Spatiotemporal Pyramidal CNN

Angga Maulana Purba [1], Fajar Mahardika [2], Satriawan Desmana [3], Nur Muniroh [1]*

[1] *Jurusan Komputer dan Bisnis, Teknologi Rekayasa Multimedia, Politeknik Negeri Cilacap, Cilacap, Indonesia*
[2] *Jurusan Komputer dan Bisnis, Teknik Informatika, Politeknik Negeri Cilacap, Cilacap, Indonesia*
[3] *Jurusan Komputer dan Bisnis, Rekayasa Keamanan Siber, Politeknik Negeri Cilacap, Cilacap, Indonesia*

INFORMASI ARTIKEL

Diterima Redaksi: 27 Oktober 2025
Revisi Akhir: 09 Januari 2025
Diterbitkan *Online*: 12 Januari 2026

KATA KUNCI

Computer Vision
Deep Learning
Driver Drowsiness Detection
Spatiotemporal

KORESPONDENSI (*)

Phone: +62 858-6762-3098
E-mail: anggamaulana@pnc.ac.id

A B S T R A K

Keselamatan berkendara merupakan hal yang penting dan menjadi topik yang krusial diperbincangkan termasuk mendeteksi kantuk pengemudi. Dalam *Computer Vision* salah satu pendekatan yang dilakukan adalah mendeteksi kedipan mata pengemudi. Penelitian tahun 2022 menunjukkan hasil yang baik dalam penggunaan *Pyramidal Bottleneck CNN* untuk mempelajari fitur spatio dan temporal pada kedipan mata. Kemudian tahun 2023 dikembangkan model yang lebih ringan dengan *Depth-wise Separable Convolution,* sehingga parameter latih bisa diperkecil. Oleh karena itu, Penelitian ini bertujuan mengadaptasi arsitektur tersebut untuk kasus keterbukaan mata. Kedipan mata berlangsung cukup singkat dan hanya sepersekian detik, sehingga belum cukup membantu untuk mendeteksi kantuk. Model tersebut berhasil dilatih pada data primer yang terbatas dan dibandingkan dengan model *baseline*. Model terbaik secara keseluruhan menghasilkan *F1 score* 0.75, *Precision* 0.63, dan *Recall* 0.93. Model tersebut bisa berjalan diatas CPU dengan rata-rata 12 FPS (*Frame Per Second*). Hasil *recall* yang cukup tinggi menunjukkan model tersebut bisa menangkap momen mata tertutup cukup banyak, hal ini cukup krusial karena kehilangan momen mata tertutup akan fatal akibatnya, meskipun harus mengorbankan *precision* atau masih tinggi *false positive*-nya.

PENDAHULUAN

Keselamatan dalam berkendara merupakan suatu hal yang penting dan masih terus dikembangkan dalam riset Computer Vision. Salah satu kunci keselamatan adalah tetap fokus dalam berkendara. Kendaraan melaju dengan sangat cepat sehingga kehilangan fokus sepersekian detik akibatnya akan sangat fatal. *The National Highway Traffic Safety Administration* memperkirakan sampai 639 kecelakaan dalam setahun disebabkan oleh rasa kantuk pada pengemudi di tahun 2022[1]. Di Indonesia ada penelitian yang menyebutkan 79% responden pernah mengantuk saat berkendara setidaknya sekali [2]. Oleh karena itu mengembangkan deteksi kantuk yang baik merupakan hal yang krusial.

Dalam Computer vision ada beberapa pendekatan dalam mendeteksi kantuk [3]. Seperti menganalisis atribut muka [4], kedipan mata, aspek rasio, dan ekspresi muka pengemudi [5]. Salah satu pendekatan yang masih aktif dalam riset adalah mendeteksi kedipan mata. Penelitian tahun 2023 menyebutkan terdapat dua tantangan yang masih menjadi perhatian [6]. Pertama, berdasarkan situasinya jarak kamera dengan wajah bisa bervariasi sehingga resolusi dan ukuran objek mata yang didapatkan menjadi bervariasi, hal ini menjadi masalah bagi model untuk mempelajari berbagai macam resolusi gambar dalam dataset. Kedua, model harus bekerja real time karena terlambat sepersekian detik saja bisa fatal akibatnya. Oleh karena itu model harus ringan dan bisa bekerja cepat dalam mendeteksi kantuk pengemudi.

https://doi.org/10.56211/sudo.v4i4.1342

Anh dkk. (2023) berusaha mengatasi tantangan-tantangan tersebut dengan mengembangakan penelitian sebelumnya yang memanfaatkan *Pyramidal Bottleneck CNN*. Sebuah spatio-temporal CNN dengan jaringan *Pyramidal Bottleneck* menunjukkan hasil yang cukup bagus dalam penelitian sebelumnya [7]. Namun model yang menggunakan *Pyramidal Bottleneck* masih menghasilkan parameter yang cukup besar untuk keperluan inferensi. Pada penelitian tersebut disebutkan model dengan performa terbaik memiliki kurang lebih 7.6 juta parameter. Oleh karena itu Anh dkk. (2023) berusaha memodifikasi arsitektur tersebut dengan memperkecil ukuran parameter Convolution layer dengan memisahkan lapisan kernel yang mempelajari tiap channel secara terpisah atau sering disebut dengan *Depth-wise separable convolution* [8]. Hasil dari penelitian tersebut menyebutkan bahwa mereka mendapatkan score *recall* yang baik dalam mendeteksi kedipan mata kiri. Sehingga memperkecil false negative atau memperkecil kehilangan momen ketika mata berkedip.

Pada implementasinya kedipan mata berlangsung sepersekian detik dan belum cukup membantu menentukan bahwa pengemudi sedang mengantuk. Oleh karena itu kontribusi penelitian ini adalah menerapkan model Anh dkk. (2023) pada kasus keterbukaan mata sebagai parameter deteksi kantuk pengemudi. Selain itu penelitian ini akan menerapkan model tersebut pada dataset primer yang lebih terbatas dan memiliki karakteristik yang mirip dengan HUST-LEBW [9]. Karakteristik tersebut yaitu diambil dari variasi *angle* berbeda dan tidak terkonstrain pada jarak, sehingga memiliki variasi gambar dengan resolusi dan ukuran yang berbeda. Kamera ditempatkan dengan berbagai variasi angle dan jarak di dalam mobil. Selanjutnya model versi ringan dari Spatio-temporal *Pyramidal Bottleneck CNN* akan dibandingkan dengan model baseline dan dilatih pada dataset primer untuk mengetahui performanya.

TINJAUAN PUSTAKA

Deteksi Kantuk Pengemudi

Deteksi kantuk pengemudi adalah proses identifikasi kondisi mengantuk atau kehilangan fokus pada seseorang yang sedang mengemudi. Kantuk merupakan salah satu faktor utama penyebab kecelakaan lalu lintas, terutama pada perjalanan jarak jauh atau malam hari. Oleh karena itu, sistem yang mampu mendeteksi tanda-tanda kantuk secara dini sangat penting untuk meningkatkan keselamatan berkendara.

Tanda-tanda kantuk yang umum dianalisis antara lain:

1. Penutupan mata dalam waktu lama (microsleep)
2. Frekuensi dan durasi kedipan
3. Posisi dan arah kepala
4. Menguap atau ekspresi wajah

Dari berbagai indikator tersebut, kondisi mata (terbuka atau tertutup) merupakan indikator visual paling umum dan mudah dianalisis secara real-time melalui pemrosesan citra/video.

Keterbukaan Mata sebagai Indikator Kantuk

Kondisi mata terbuka dan tertutup menjadi parameter utama dalam banyak sistem deteksi kantuk. Ketika seseorang mengantuk, biasanya mata cenderung lebih sering tertutup, atau durasi kedipan menjadi lebih panjang. Beberapa indikator yang digunakan dalam mendeteksi kondisi mata antara lain:

1. Eye Aspect Ratio (EAR)
   Mengukur rasio tinggi dan lebar mata dari landmark wajah. Rasio yang menurun menandakan mata tertutup.
2. Percentage of Eye Closure (PERCLOS)
   Menghitung persentase waktu mata tertutup dalam interval tertentu.
3. Durasi dan frekuensi kedipan
   Orang yang mengantuk biasanya memiliki kedipan lambat dan lebih jarang dibanding kondisi sadar.

Deteksi keterbukaan mata dapat dilakukan melalui analisis urutan citra/video menggunakan teknik deteksi wajah dan ekstraksi area mata, lalu diklasifikasikan sebagai terbuka atau tertutup.

Convolutional Neural Network (CNN)

CNN adalah arsitektur deep learning yang sangat efektif dalam pengolahan citra. CNN bekerja dengan mengenali pola visual seperti tepi, bentuk, dan tekstur dari gambar melalui proses konvolusi. Komponen utama dalam CNN adalah:

1. Convolutional Layer
   Menerapkan filter untuk mengekstrak fitur.

2. Pooling Layer
   Mengurangi dimensi fitur sambil mempertahankan informasi penting.
3. Fully Connected Layer
   Mengklasifikasikan fitur menjadi output (misalnya: mata terbuka vs tertutup). CNN biasa digunakan dalam deteksi objek, pengenalan wajah, dan klasifikasi citra medis — dan telah terbukti efektif dalam mengenali kondisi mata dari citra wajah.

Research Gap pada Deteksi Kantuk

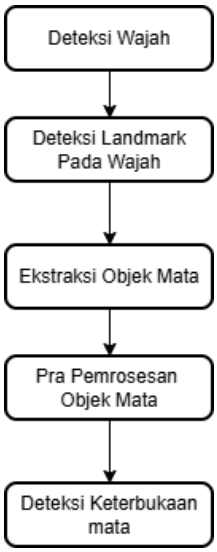
Penelitian sebelumnya mengenai deteksi kantuk umumnya berupa analisis wajah, mata, atau kombinasi beberapa fitur dari area wajah. Metode klasik seperti *Histogram Of Oriented Gradients* juga sering dipakai untuk mengekstrasi fitur teksture pada wajah. Namun metode tersebut memerlukan pra pemrosesan yang cukup kompleks sebelum dilakukan klasifikasi akhir. Oleh karena itu pendekatan menggunakan Neural Network lebih diminati karena memiliki potensi optimisasi pada perangkat *edge devices.* Penelitian berbasis CNN masih terus dikembangkan seperti terlihat pada Tabel I. Berdasarkan tabel tersebut masih ada celah untuk menguji model *Depth-wise separable Pyramidal CNN* pada kasus keterbukaan mata pada dataset yang lebih terbatas.

Tabel 1. Penelitian Deteksi Kantuk

Peneliti	Pendekatan	Metode	Dataset	Tahun
S. Bakheet dkk. [4]	Analisis Wajah	HOG	NTHU-DDD	2021
Md. T. A. Dipu dkk. [5]	Keterbukaan mata	mobilenet	custom	2021
S. E. Bekhouche dkk. [7]	Kedipan Mata	Pyramidal CNN	HUST-LEBW	2022
Anh dkk. [6]	Kedipan Mata	Depth-wise separable convolution Pyramidal CNN	HUST-LEBW	2023
Penelitian ini	Keterbukaan Mata	Perbandingan Pyramidal CNN dan depthwise	custom	2025

METODOLOGI

Penelitian ini akan menggunakan pendekatan kuantitatif eksperimental untuk menguji performa dari model. Secara garis besar proses deteksi kantuk dijelaskan pada Gambar 1. Deteksi wajah dan objek mata menggunakan library mediapipe yang tersedia secara opensource [10]. Library tersebut dipilih karena kecepatannya dan penggunaan resource yang tidak terlalu berat [11]. Setelah objek mata didapatkan selanjutnya deteksi keterbukaan mata dilakukan dengan satu referensi saja yaitu mata kiri. Hal itu dilakukan karena sebagian besar kasus mata memiliki karakteristik yang simetris dan mempunyai behaviour yang sama baik mata kanan ataupun kiri. Selain itu menurut rumus EAR (Eye Aspect Ratio), salah satu mata dapat menentukan derajat keterbukaan mata [12].



Gambar 1. Diagram alur Deteksi Keterbukaan mata

Pengumpulan Data

Data yang dikumpulkan yaitu berupa data primer dan tambahan data sintesis. Data primer diambil dari 2 angle kamera berbeda di dalam mobil yaitu posisi depan dan posisi sedikit samping dari pengemudi. Data diambil dengan berbagai variasi cahaya yang ada didalam mobil. Contoh pengambilan angle kamera dapat dilihat di Gambar 2. Sedangkan data sintesis diambil dari berbagai sumber terbuka di internet untuk menambahkan variasi dataset.

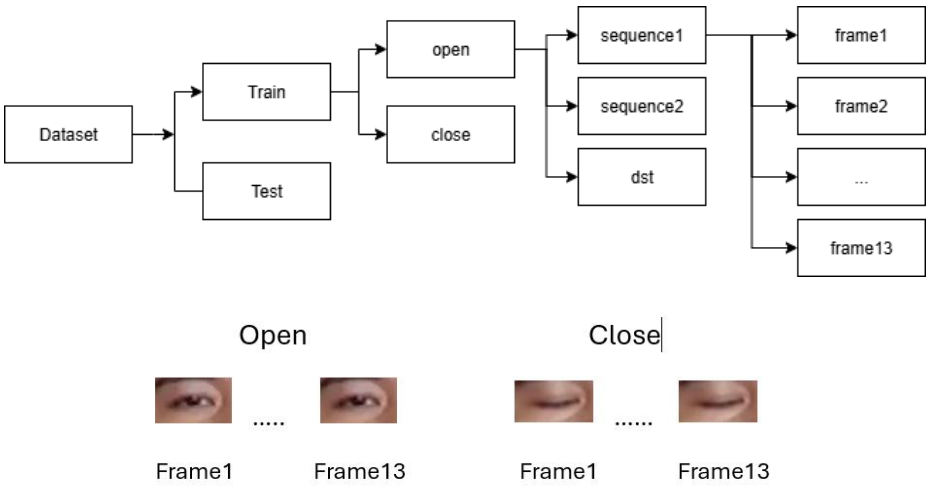
Gambar 2. Posisi Pengambilan Gambar

Data primer terdiri dari 4 pengemudi yang berbeda. Tinggi masing-masing pengemudi bervariasi sehingga menghasilkan posisi duduk didalam mobil yang berbeda. Salah satu pengemudi juga menggunakan kacamata untuk menambahkan variasi dalam dataset mata yang akan dikumpulkan. Variasi tersebut menghasilkan ukuran mata yang berbeda-beda.

Data objek mata diambil menggunakan library mediapipe yang berhasil mendeteksi mata kiri dengan baik dalam berbagai kondisi. Kemudian data diambil berupa urutan frame berjumlah 13 frame sebagai fitur temporal. Data ini yang nantinya dimasukkan ke dalam 3D tensor. Kemudian masing-masing diberikan label untuk menandakan mata sedang terbuka dan tertutup. Keseluruhan data yang dihasilkan dapat dilihat pada tabel II dan struktur dataset dapat dilihat di Gambar 3.

Tabel 2. Keseluruhan Dataset

Label	Train	Test
open	176	132
closed	145	218



Gambar 3. Struktur Folder Dataset dan Contoh Dataset

Pemrosesan Data

Model yang digunakan akan menerima input berupa 3D tensor dengan urutan frame sebagai fitur temporal dan gambar 3 channel dengan ukuran 96x96 sebagai fitur spatial. Sebelum dimasukkan ke dalam model terlebih dahulu dilakukan resizing gambar ke ukuran 96x96 menggunakan opencv. Pada penelitian Anh dkk. (2023) menyebutkan pengaruh algoritma interpolation terhadap performa model. Sehingga gambar dengan ukuran kurang dari 96 dilakukan algoritma bicubic (INTER_CUBIC). Sedangkan yang lebih dari 96 dilakukan resizing dengan algoritma bilinear (INTER_LINEAR).

 https://doi.org/10.56211/sudo.v4i4.1342

Pelatihan Model dan Evaluasi

Pelatihan model menggunakan metode stratified K-Fold yaitu metode yang digunakan untuk membagi dataset training kedalam sejumlah bagian. Satu bagian sebagai data validasi dan lainnya sebagai data training. Perbedaan K-Fold biasa dengan stratified K-fold adalah data dibagi secara proporsional terhadap label yang ada [13]. Kemudian Model terbaik akan dievaluasi terhadap data testing menggunakan beberapa metrik yaitu Precision, Recall dan F1.

1. Precision
   Precision adalah metrik untuk mengukur kemampuan model dalam mendeteksi *true positive*. Metrik ini sering digunakan untuk mengurangi *false positive* sehingga model akan mendeteksi dengan benar apakah mata benar-benar tertutup [14].

$$P = \frac{\text{TP}}{\text{TP+FP}} \qquad (1)$$

2. Recall
   Recall adalah metrik yang digunakan untuk mengukur rasio *true positive* terhadap objek-objek yang relevan. Dalam konteks deteksi kantuk dapat digunakan untuk memperkecil kehilangan momen saat mata sedang tertutup [15].

$$R = \frac{\text{TP}}{\text{TP+FN}} \qquad (2)$$

3. F1 Score
   Nilai metrik F1 digunakan untuk melihat keseimbangan antara *Precision* dan *Recall* [16].

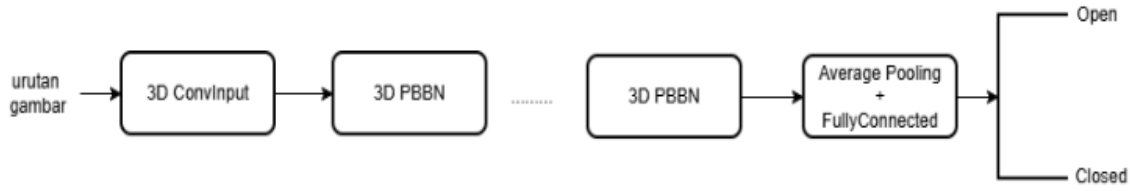
$$F1 = \frac{2 \text{ x P x R}}{P+R} \qquad (3)$$

HASIL DAN PEMBAHASAN

Model yang akan digunakan untuk mempelajari fitur spatio dan temporal adalah *Pyramid Bottleneck Block Network (PBBN) baseline* dan juga PBBN hasil modifikasi Anh dkk.(2023). Pada penelitian Bekhouche dkk. (2022) arsitektur ini digunakan untuk mempelajari berbagai resolusi dari input dari sebuah 3D tensor. Dalam konteks deteksi kantuk dari masukan berupa urutan frame, dapat diinterpretasikan sebagai dua fitur spatial (Panjang dan lebar) dan urutan frame sebagai fitur temporal.

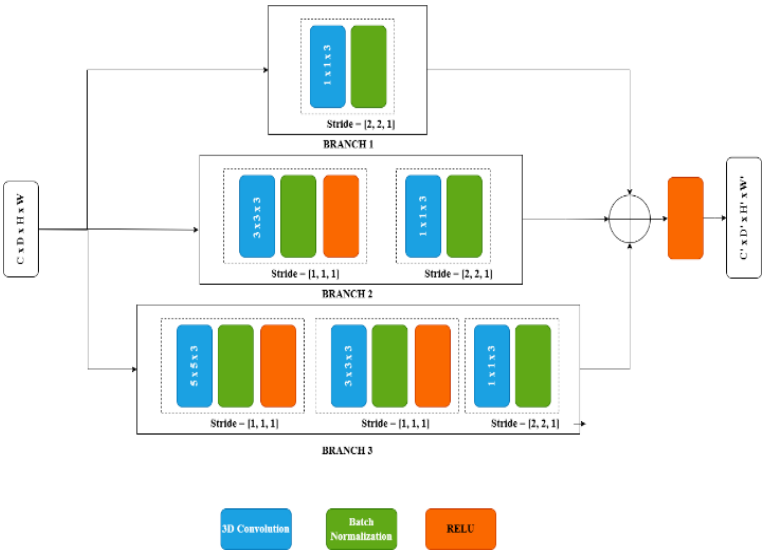
Arsitektur 3D PBBN baseline dan Depth Wise 3D PBBN

Secara garis besar 3D PBBN di gambarkan pada Gambar 4. Model menerima masukkan 3D Convolution Input kemudian diikuti beberapa 3D PBBN. Hasilnya dilakukan average pooling yang terhubung ke dua output secara *fully connected* yang diikuti layer *softmax* untuk hasil akhir.

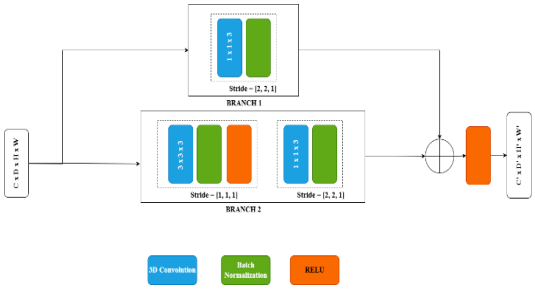


Gambar 4. Garis Besar arsitektur 3D PBBN

Pada penelitian ini akan digunakan 3D PBBN dengan jumlah variasi jumlah pyramid dan cabang yaitu 2 piramid 2 cabang (P2B2) dan 3 piramid 3 cabang (P3B3). Secara garis besar arsitektur tersebut dapat digambarkan pada Gambar 5 dan 6. Kita asumsikan urutan cabang dengan symbol *l,* maka cabang ke *l* diawali oleh convolution layer dengan ukuran filter (2*l*-1) x (2*l*-1) lalu semakin mengecil setengahnya pada cabang yang sama. Contoh branch 2 diawali oleh 3x3x3 kemudian mengecil ke 1x1x3, hal sama juga dilakukan pada cabang ke 3.



Gambar 5. 3D PBBN dengan 3 cabang



Gambar 6. 3D PBBN dengan 2 cabang

Arsitektur 3D PBBN tersebut menghasilkan model yang cukup dalam untuk mempelajari input dengan beberapa resolusi. Namun sebagai gantinya model tersebut memiliki jumlah parameter yang cukup besar untuk dilatih. Perbandingan secara jelas antara 3D PBBN baseline 2 piramid dan 2 cabang dengan versi modifikasi *depthwise separable convolution* digambarkan pada Tabel 2 dan 3. Perbandingan total parameter hampir 8 kali lipatnya. Hal ini tentu akan berdampak pada waktu dan resource yang digunakan baik saat melakukan training maupun inferensi saat diimplementasikan.

Tabel 2. Arsitektur 3D PBBN dengan 2 Piramid dan 2 Cabang

type	out_shape	params
Conv3d	(1, 64, 7, 96, 96)	5,248
BatchNorm3d	(1, 64, 7, 96, 96)	128
ReLU	(1, 64, 7, 96, 96)	0
MaxPool3d	(1, 64, 4, 96, 96)	0
Conv3d	(1, 64, 4, 48, 48)	12,352
BatchNorm3d	(1, 64, 4, 48, 48)	128
Conv3d	(1, 64, 4, 96, 96)	110,656
BatchNorm3d	(1, 64, 4, 96, 96)	128
ReLU	(1, 64, 4, 96, 96)	0
Conv3d	(1, 64, 4, 48, 48)	12,352
BatchNorm3d	(1, 64, 4, 48, 48)	128

 https://doi.org/10.56211/sudo.v4i4.1342

ReLU	(1, 64, 4, 48, 48)	0
Conv3d	(1, 128, 4, 24, 24)	24,704
BatchNorm3d	(1, 128, 4, 24, 24)	256
Conv3d	(1, 128, 4, 48, 48)	221,312
BatchNorm3d	(1, 128, 4, 48, 48)	256
ReLU	(1, 128, 4, 48, 48)	0
Conv3d	(1, 128, 4, 24, 24)	49,280
BatchNorm3d	(1, 128, 4, 24, 24)	256
ReLU	(1, 128, 4, 24, 24)	0
AvgPool3d	(1, 128, 1, 1, 1)	0
Linear	(2,)	258
Total Parameter		437,442

Tabel 3. Arsitektur *Depthwise Separable Conv 3D PBBN* dengan 2 Piramid 2 Cabang

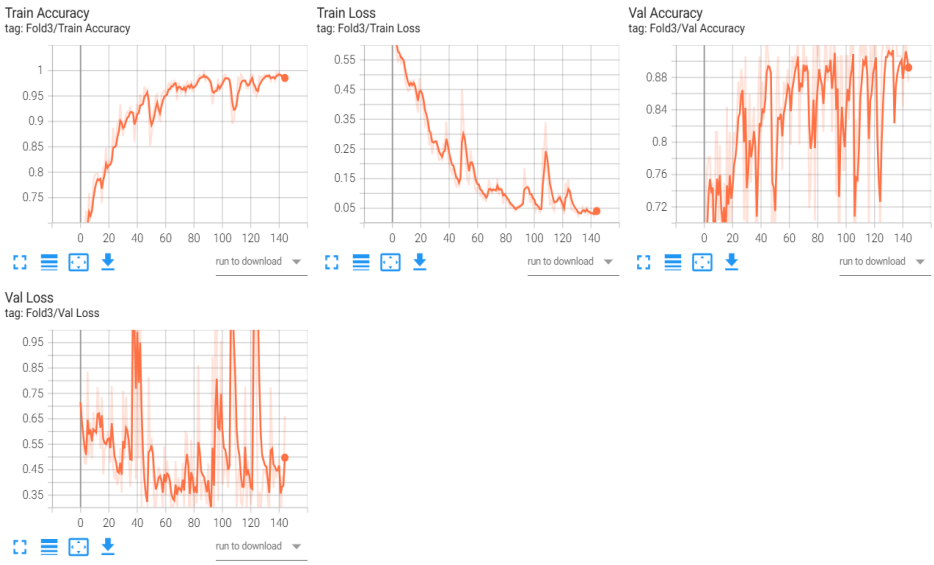
type	out_shape	params
Conv3d	(1, 64, 7, 96, 96)	5,248
BatchNorm3d	(1, 64, 7, 96, 96)	128
ReLU	(1, 64, 7, 96, 96)	0
MaxPool3d	(1, 64, 4, 96, 96)	0
Conv3d	(1, 64, 4, 48, 48)	256
BatchNorm3d	(1, 64, 4, 48, 48)	128
ReLU	(1, 64, 4, 48, 48)	0
Conv3d	(1, 64, 4, 48, 48)	4,160
BatchNorm3d	(1, 64, 4, 48, 48)	128
Conv3d	(1, 64, 4, 96, 96)	1,792
BatchNorm3d	(1, 64, 4, 96, 96)	128
ReLU	(1, 64, 4, 96, 96)	0
Conv3d	(1, 64, 4, 96, 96)	4,160
BatchNorm3d	(1, 64, 4, 96, 96)	128
ReLU	(1, 64, 4, 96, 96)	0
Conv3d	(1, 64, 4, 48, 48)	256
BatchNorm3d	(1, 64, 4, 48, 48)	128
ReLU	(1, 64, 4, 48, 48)	0
Conv3d	(1, 64, 4, 48, 48)	4,160
BatchNorm3d	(1, 64, 4, 48, 48)	128
ReLU	(1, 64, 4, 48, 48)	0
Conv3d	(1, 64, 4, 24, 24)	256
BatchNorm3d	(1, 64, 4, 24, 24)	128
ReLU	(1, 64, 4, 24, 24)	0
Conv3d	(1, 128, 4, 24, 24)	8,320
BatchNorm3d	(1, 128, 4, 24, 24)	256
Conv3d	(1, 64, 4, 48, 48)	1,792
BatchNorm3d	(1, 64, 4, 48, 48)	128
ReLU	(1, 64, 4, 48, 48)	0
Conv3d	(1, 128, 4, 48, 48)	8,320
BatchNorm3d	(1, 128, 4, 48, 48)	256
ReLU	(1, 128, 4, 48, 48)	0
Conv3d	(1, 128, 4, 24, 24)	512
BatchNorm3d	(1, 128, 4, 24, 24)	256
ReLU	(1, 128, 4, 24, 24)	0

Conv3d	(1, 128, 4, 24, 24)	16,512
BatchNorm3d	(1, 128, 4, 24, 24)	256
ReLU	(1, 128, 4, 24, 24)	0
AvgPool3d	(1, 128, 1, 1, 1)	0
Linear	(2,)	258
Total Parameter		58,178

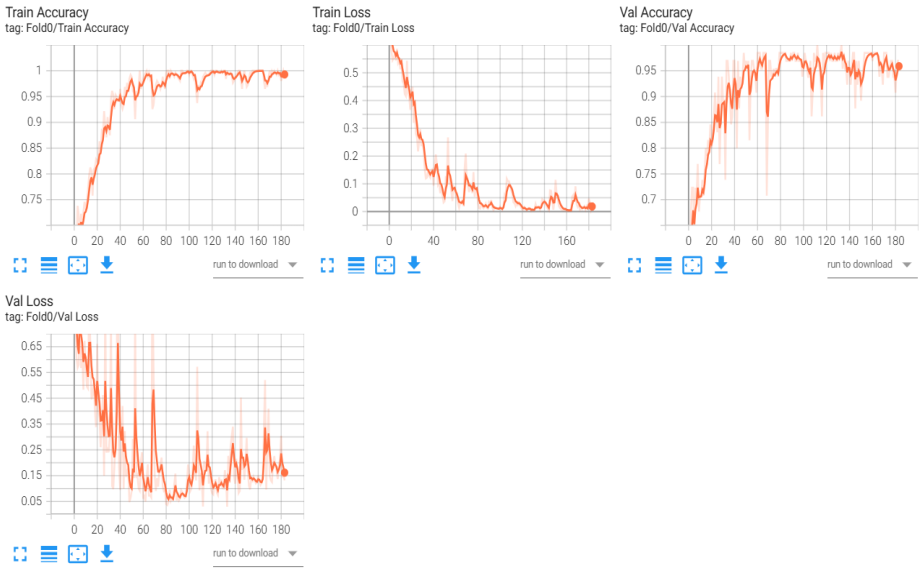
Pelatihan Model dan Validasi

3D PBBN baseline

Model dilatih dengan dengan menggunakan GPU P100 milik Kaggle. Dataset dibagi menjadi 5 fold dan Masing-masing fold menjalankan 200 epoch. Model 3D PBBN dengan 2 piramid dan 2 cabang menghabiskan waktu training 4 jam 45 menit, sedangkan 3 Piramid dan 3 cabang menghabiskan waktu 12 jam 15 menit. Hal yang perlu diperhatikan adalah potensi overfitting pada model, terlihat dari *validation loss* dan *Accuracy* yang fluktuatif meskipun *train accuracy* sudah sangat tinggi. Oleh karena itu, dilakukan pemberhentian lebih awal saat epoch berjalan [17]. Terlihat pada gambar epoch berhenti sebelum menuju 200 karena model tidak berhasil memperkecil *validation loss*. Saat *validation loss* tidak berkurang sampai 50 langkah maka epoch berhenti.



Gambar 7. Training dan Validation pada 2 Piramid dengan 2 cabang pada Fold terbaik

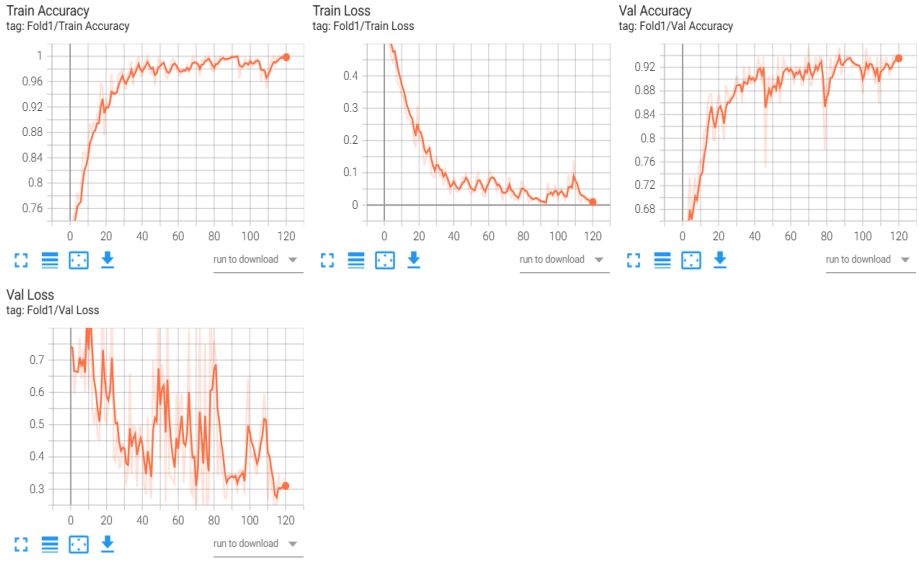


Gambar 8. Training dan Validation pada 3 pyramid dengan 3 cabang pada fold terbaik

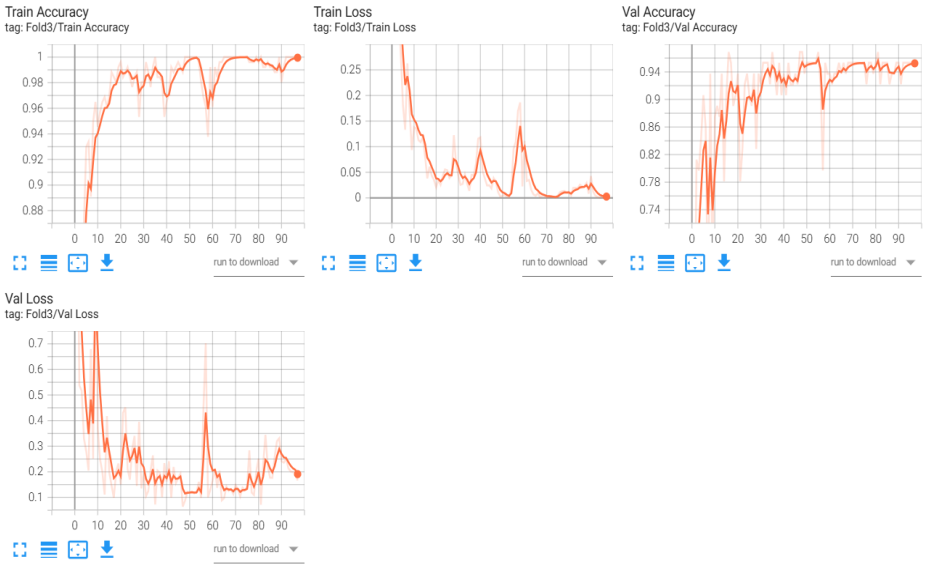
 https://doi.org/10.56211/sudo.v4i4.1342

Depthwise Separable Convolution Layer 3D PBBN

Model dilatih dengan hardware dan perlakuan yang sama pada model baseline. Didapatkan hasil yang tidak begitu berbeda dengan baseline pada tahap validasi ini meskipun piramid ditambahkan di arsitektur. Sebagai gantinya waktu training jauh lebih singkat yaitu 1.5 jam untuk 2 piramid dan 2 cabang sedangkan untuk 3 piramid dan 3 cabang membutuhkan waktu sekitar 2 jam 13 menit. Dari grafik gambar terlihat model yang lebih kecil dari baseline ternyata lebih baik dalam menangani *overfitting*. Terlihat pada grafik model lebih cepat mendapatkan konvergensi pada *validation loss* dengan epoch lebih sedikit daripada model baseline.



Gambar 9. Training dan Validation pada 2 piramid dengan 2 cabang pada fold terbaik



Gambar 10. Taining dan Validation pada 3 pyramid dan 3 cabang pada fold terbaik

Testing Model dan Evaluasi

Setelah tahap validasi model terbaik dapat dipilih dari 5 fold yang dilatih dan dapat dilakukan evaluasi terhadap data testing. Pemisahan validasi dan testing ini memastikan model dapat belajar pada data yang belum pernah dilihat sama sekali. Model akan dievaluasi terhadap data testing berjumlah 363 dengan label *closed* sejumlah 145 dan label *open* sejumlah 218. Untuk keperluan kecepatan inferensi, model akan diuji menggunakan hardware berupa Laptop dengan CPU intel Core i5 generasi ke 12 dan GPU RTX 3050 dengan memory 6 GB.

Pada Tabel 4 terlihat bahwa model 3D PBBN baseline dengan 3 piramid dan 3 cabang (P3B3) memiliki precision yang terbaik. Hal ini menggambarkan model ini memiliki Tingkat *false positive* terendah dan tingkat kebenaran yang tinggi. Sedangkan versi yang lebih ringan memiliki *recall* yang lebih besar sehingga lebih banyak momen mata tertutup berhasil ditangkap oleh model ini. Secara keseluruhan metrik model dengan 3 piramid dan 3 cabang (P3B3) memberikan performa terbaik secara *Precision, Recall dan F1*. Hanya saja perlu dipertimbangkan hal-hal lain dalam mengukur performa keseluruhan seperti waktu dan resource saat training serta performa saat inferensi dengan hardware yang terbatas.

Tabel 4 Metrik Masing-masing Model

Model	F1	Precision	Recall
P2B2	0.765273	0.664804	0.901515
P2B2 Separable Conv	0.752294	0.630769	0.931818
P3B3	0.879999	**0.846153**	0.916666
P3B3 Separable Conv	**0.902098**	0.837662	**0.977273**

Pada Tabel 4 model dengan 2 piramid memberikan kecepatan inferensi yang lebih baik dengan recall yang masih diatas 90%. Hal ini menunjukkan model masih memiliki kemampuan yang baik menangkap momen mata tertutup. Sesuai dengan penelitian sebelumnya baik model baseline maupun *depthwise version* memiliki tingkat *recall* yang cukup baik. Hal itu bisa terjadi karena model *pyramid block* dapat mempelajari fitur spatio-temporal pada resolusi yang berbeda tanpa memerlukan arsitektur yang lebih dalam [6]. Meskipun begitu model memiliki *precision* yang tidak terlalu baik sehingga tingkat *false positif* masih cukup tinggi. Hal ini mungkin dipengaruhi oleh dataset yang terbatas dan juga tidak seimbang. Pada penelitian sebelumnya dengan dataset HUBSET-LEBW, Anh dkk. (2023) juga menghadapi masalah yang sama terkait *precision* yang menurun akibat arsitektur *depthwise* dibandingkan dengan *baseline*. Sehingga hasil pengujian pada dataset *custom* penelitian ini merefleksikan hasil penelitian pada kasus kedipan mata sebelumnya.

Dalam Implementasinya model dengan P2B2 masih bekerja cukup baik dengan segala kekurangan tersebut. Terlihat pada Tabel 4 model dengan 2 piramida menghasilkan rata-rata FPS (*Frame Per Second*) tertinggi saat dijalankan diatas CPU. Meskipun rerata terbaik yang dihasilkan hanya 12 FPS. Tabel 5 juga menunjukkan bahwa model 2 piramid memiliki waktu training yang paling cepat diantara model-model yang lain. Hal tersebut membuktikan modifikasi *depthwise separable convolution* sangat berpengaruh mereduksi waktu training cukup signifikan.

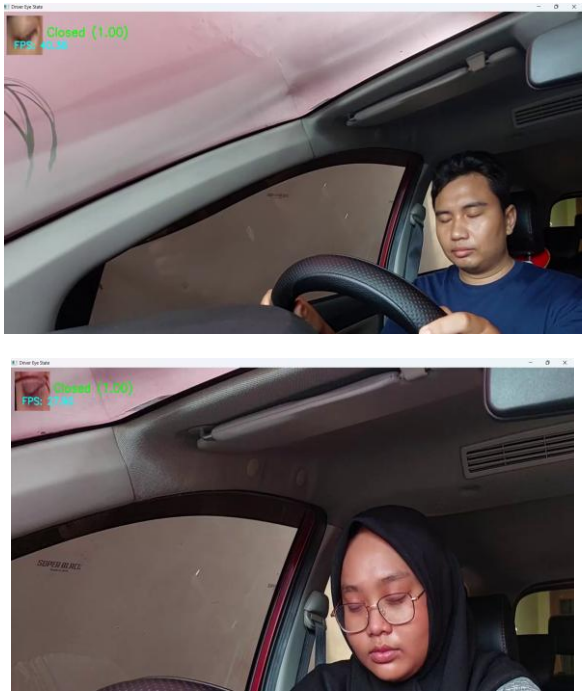
Tabel 5 Rerata FPS pada CPU

Model	Rata-rata FPS
P2B2	8
P2B2 Separable Conv	**12**
P3B3	3
P3B3 Separable Conv	7

Tabel 6 Waktu Training

Model	Waktu Training
P2B2	4 jam 45 menit
P2B2 Separable Conv	**1 jam 30 menit**
P3B3	12 jam 15 menit
P3B3 Separable Conv	2 jam 1 menit

https://doi.org/10.56211/sudo.v4i4.1342



Gambar 11 Contoh Inferensi dengan model P2B2 Separable Conv yang berjalan diatas GPU

KESIMPULAN DAN SARAN

Penelitian ini menunjukkan bahwa arsitektur *Pyramid Bottleneck Block Network* (PBBN) baik *baseline* maupun *Depthwise version* dapat bekerja cukup baik pada kasus deteksi kantuk dengan keterbukaan mata. Hasil menunjukkan model berhasil mencapai F1 score yang cukup tinggi. Hal ini menunjukkan bahwa model bekerja cukup bagus pada dataset yang terbatas dan tidak seimbang dimana akurasi tidak bisa dijadikan pedoman. Selain itu model juga memiliki Tingkat *recall* yang cukup bagus, hal ini cukup penting karena sebisa mungkin model harus bisa menangkap momen mata tertutup pengemudi sebanyak mungkin. Lebih baik mengorbankan *precision* karena kehilangan momen mata tertutup akan sangat fatal akibatnya. Meskipun begitu model masih cukup berat untuk dijalankan pada hardware yang terbatas dan hanya menghasilkan rata-rata 12 FPS pada model paling ringan. Saran untuk penelitian selanjutnya adalah *benchmarking* model ke perangkat tepi di pasaran seperti jetson nano, raspberry atau orangepi. Selain itu bisa dilakukan *ablation study* dan perbandingan dengan model *lightweight* yang lain pada dataset yang lebih besar.

DAFTAR PUSTAKA

[1] National Center for Statistics and Analysis, “Overview of Motor Vehicle Traffic Crashes in 2022,” Washington, D.C., 2024.

[2] R. Rahmadiyani and A. widyanti, “Prevalence of drowsy driving and modeling its intention: An Indonesian case study,” *Transp Res Interdiscip Perspect*, vol. 19, p. 100824, May 2023, doi: 10.1016/j.trip.2023.100824.

[3] B. Fu, F. Boutros, C.-T. Lin, and N. Damer, “A Survey on Drowsiness Detection -- Modern Applications and Methods,” Aug. 2024.

[4] S. Bakheet and A. Al-Hamadi, “A Framework for Instantaneous Driver Drowsiness Detection Based on Improved HOG Features and Naïve Bayesian Classification,” *Brain Sci*, vol. 11, no. 2, p. 240, Feb. 2021, doi: 10.3390/brainsci11020240.

[5] Md. T. A. Dipu, S. S. Hossain, Y. Arafat, and F. B. Rafiq, “Real-time Driver Drowsiness Detection using Deep Learning,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, 2021, doi: 10.14569/IJACSA.2021.0120794.

[6] N. T. L. Anh, N. G. Bach, N. T. T. Tu, E. Kamioka, and P. X. Tan, “Spatiotemporal Pyramidal CNN with Depth-Wise Separable Convolution for Eye Blinking Detection in the Wild,” *Journal of Image and Graphics*, vol. 11, no. 4, pp. 367–375, Dec. 2023, doi: 10.18178/joig.11.4.367-375.

[7] S. E. Bekhouche, I. Kajo, Y. Ruichek, and F. Dornaika, "Spatiotemporal CNN with Pyramid Bottleneck Blocks: Application to eye blinking detection," *Neural Networks*, vol. 152, pp. 150–159, Aug. 2022, doi: 10.1016/j.neunet.2022.04.010.

[8] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," Apr. 2017.

[9] G. Hu *et al.*, "Towards Real-time Eyeblink Detection in The Wild:Dataset,Theory and Practices," Dec. 2019, doi: 10.1109/TIFS.2019.29599778.

[10] C. Lugaresi *et al.*, "MediaPipe: A Framework for Building Perception Pipelines," Jun. 2019.

[11] A. G. Sawant, S. S. Kamble, R. S. Kanade, R. N. Kanugo, T. A. Kapse, and K. A. Bhapse, "A Real-Time Driver Drowsiness Detection System Using MediaPipe and Eye Aspect Ratio," Nov. 17, 2025, arXiv: arXiv:2511.13618. doi: 10.48550/arXiv.2511.13618.

[12] F. Safarov, F. Akhmedov, A. B. Abdusalomov, R. Nasimov, and Y. I. Cho, "Real-Time Deep Learning-Based Drowsiness Detection: Leveraging Computer-Vision and Eye-Blink Analyses for Enhanced Road Safety," Sensors, vol. 23, no. 14, p. 6459, Jul. 2023, doi: 10.3390/s23146459.

[13] S. Prusty, S. Patnaik, and S. K. Dash, "SKCV: Stratified K-fold cross-validation on ML classifiers for predicting cervical cancer," *Frontiers in Nanotechnology*, vol. 4, Aug. 2022, doi: 10.3389/fnano.2022.972421.

[14] M. Fauzan Ridho, Fransiskus Panca, Welly Yandi, and Almeera Amsana Rachmani, "Drowsiness Detection in the Advanced Driver-Assistance System using YOLO V5 Detection Model," *ELECTRON Jurnal Ilmiah Teknik Elektro*, vol. 5, no. 1, May 2024, doi: 10.33019/electron.v5i1.136.

[15] Z. Wang, K. Yao, and F. Guo, "Driver Attention Detection Based on Improved YOLOv5," *Applied Sciences*, vol. 13, no. 11, p. 6645, May 2023, doi: 10.3390/app13116645.

[16] S. LIU, Y. WANG, Q. YU, J. ZHAN, H. LIU, and J. LIU, "A Driver Fatigue Detection Algorithm Based on Dynamic Tracking of Small Facial Targets Using YOLOv7," *IEICE Trans Inf Syst*, vol. E106.D, no. 11, p. 2023EDP7093, Nov. 2023, doi: 10.1587/transinf.2023EDP7093.

[17] K. Anam, "Early Stopping on CNN-LSTM Development to Improve Classification Performance," *Journal of Applied Data Sciences*, vol. 5, no. 3, pp. 1175–1188, Sep. 2024, doi: 10.47738/jads.v5i3.312.