

Penerapan Arsitektur MobileNetV2 Berbasis Replikasi Channel untuk Klasifikasi Citra Grayscale

Miranda Dwi Diana¹, Yuni Yamasari²,

^{1,2} Program Studi S1 Teknik Informatika, Universitas Negeri Surabaya

[1miranda.22117@mhs.unesa.ac.id](mailto:miranda.22117@mhs.unesa.ac.id)

[2yuniyamasari@unesa.ac.id](mailto:yuniyamasari@unesa.ac.id)

Abstrak—Kecelakaan lalu lintas masih menjadi permasalahan serius yang berdampak pada tingginya angka cedera dan kematian, di mana salah satu penyebab utamanya adalah kondisi mengantuk yang menurunkan tingkat konsentrasi pengemudi. Oleh karena itu, diperlukan suatu sistem deteksi mata tertutup dan terbuka yang akurat dan efisien. Pendekatan berbasis citra mata menjadi solusi yang potensial karena mampu mengidentifikasi kondisi mata terbuka dan tertutup secara otomatis menggunakan metode pengolahan citra dan deep learning. Penggunaan citra *grayscale* dinilai lebih efisien dari segi komputasi, namun memiliki keterbatasan karena tidak sesuai dengan arsitektur model pre-trained yang umumnya memerlukan input tiga channel. Penelitian ini bertujuan untuk menerapkan arsitektur *MobileNetV2* berbasis replikasi channel dalam klasifikasi citra grayscale untuk deteksi mata tertutup dan terbuka. Teknik replikasi channel dilakukan dengan menggandakan citra *grayscale* menjadi tiga channel agar kompatibel dengan arsitektur *MobileNetV2*. Model dikembangkan menggunakan pendekatan *transfer learning* untuk meningkatkan kemampuan ekstraksi fitur dan performa klasifikasi. Hasil pengujian menunjukkan bahwa model *MobileNetV2* dengan teknik replikasi channel mampu mencapai akurasi sebesar 99,92% dalam mengklasifikasikan kondisi mata terbuka dan tertutup. Hal ini menunjukkan bahwa metode yang diusulkan efektif dalam meningkatkan kompatibilitas citra *grayscale* dengan model *deep learning pre-trained* serta mampu menghasilkan performa klasifikasi yang tinggi.

Kata Kunci—deteksi kelelahan, *MobileNetV2*, replikasi channel, citra *grayscale*, *transfer learning*.

I. PENDAHULUAN

Kecelakaan lalu lintas menjadi salah satu penyebab utama kematian di seluruh dunia. Menurut data WHO, sekitar 1,3 juta orang meninggal dunia setiap tahun akibat kecelakaan lalu lintas[1]. Kelelahan pengemudi merupakan faktor risiko signifikan dalam kecelakaan lalu lintas, yang berkontribusi terhadap angka kematian yang tinggi di seluruh dunia. NHTSA (*National Highway Traffic Safety Administration*) memperkirakan bahwa pada tahun 2017, mengemudi dalam keadaan mengantuk menyebabkan 91.000 kecelakaan[2]. Berdasarkan Fakta tersebut menunjukkan bahwa perlu adanya perhatian khusus dalam mengurangi kasus kecelakaan tersebut. Dua faktor utama penyebab kecelakaan lalu lintas adalah kelelahan pengemudi dan gangguan konsentrasi. Organisasi Kesehatan Dunia (WHO) melaporkan bahwa 7% kecelakaan lalu lintas fatal dan parah disebabkan oleh mengemudi dalam keadaan mengantuk. Kondisi kantuk berpengaruh pada konsentrasi dan waktu reaksi pengemudi, sehingga sistem deteksi sangat diperlukan [3]. Dengan adanya kemajuan yang signifikan dalam bidang penelitian dan teknologi. Sistem yang dibangun dapat digunakan untuk

melindungi nyawa pengemudi, terutama dengan penggunaan teknologi berbasis gambar. Salah satu metode yang sering digunakan yaitu *MobileNetV2* untuk deteksi kelelahan pengemudi[4].

Tujuan utamanya penelitian ini adalah mengembangkan sistem yang fleksibel dan efektif yang berkinerja baik, mengurangi risiko kelelahan pengemudi, dan meningkatkan keselamatan lalu lintas. Tidak seperti teknik sebelumnya, yang seringkali bergantung pada algoritma dasar atau strategi yang lugas, penelitian ini memanfaatkan beberapa arsitektur dengan metode deep learning untuk menyelesaikan masalah tersebut. Dalam Mendeteksi dan mencegah kelelahan saat mengemudi, berbagai pendekatan telah dikembangkan yang menargetkan dan mengukur kinerja pengemudi, perilaku, dan kondisi fisiologis. Pada penelitian ini berfokus pada kondisi mata pengemudi. Dengan mengklasifikasi kondisi mata tertutup atau mata terbuka pengemudi[5]

Banyak penelitian terdahulu yang telah memanfaatkan CNN untuk mendeteksi kelelahan pada citra pengemudi. Menerapkan transfer learning *MobileNet* untuk mengklasifikasi status mata (terbuka/tertutup) pada video pengemudi. Model berbasis *MobileNet* yang mereka latih berhasil mencapai akurasi sekitar 90%[6]. Secara keseluruhan, kajian pustaka menunjukkan bahwa *MobileNetV2* telah berhasil digunakan untuk deteksi kelelahan berbasis citra mata. Penerapan *deep learning*, khususnya arsitektur transfer learning seperti arsitektur *MobileNetV2*, mampu mendeteksi kelelahan pengemudi dengan akurasi tinggi[7]. Dalam kondisi nyata pengambilan gambar yang menghasilkan citra *grayscale* untuk kondisi pencahayaan rendah atau sederhana. Sebagian besar CNN *pre-trained* memerlukan input tiga saluran (RGB), sehingga citra *grayscale* perlu diubah agar sesuai. Sebuah teknik yang umum dipakai adalah replikasi channel menyalin satu saluran *grayscale* menjadi tiga saluran *pseudo-RGB* yang identik[8].

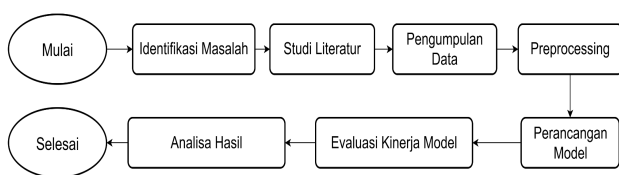
Dataset citra mata pengemudi juga berkembang pesat. Misalnya, *MRL Eye Dataset* yang tersedia di Kaggle memiliki 84.898 citra mata terbuka/tertutup[9]. Namun, pelatihan CNN membutuhkan sumber daya komputasi besar, sehingga pada penelitian ini hanya digunakan subset sebanyak 4.000 citra dari *MRL Eye Dataset* sebagai sampel pelatihan dan pengujian. Pendekatan ini menekankan bahwa hanya 4.000 citra digunakan demi keterbatasan komputasi. Selain itu, karena citra dalam MRL berformat *grayscale* satu kanal, perlu dilakukan replikasi *channel* agar sesuai input arsitektur *MobileNetV2* berprasetel yang umumnya memerlukan tiga

kanal. Dalam proses pengolahan citra di-resize dan replikasi agar model lebih optimal[10].

Berdasarkan permasalahan yang ada, diperlukan penelitian yang mengevaluasi penerapan arsitektur *MobileNetV2* untuk klasifikasi citra mata grayscale dengan teknik replikasi channel. Penelitian ini melakukan penerapan arsitektur *MobileNetV2* berbasis replikasi channel untuk mendeteksi kelelahan melalui citra mata pengemudi. arsitektur *MobileNetV2* akan dievaluasi untuk menentukan keakuratannya dalam konteks tersebut.

II. METODOLOGI PENELITIAN

Metodologi penelitian merupakan kerangka pedoman yang digunakan dalam pelaksanaan penelitian untuk memastikan setiap tahapan dilakukan secara sistematis, terstruktur, dan terarah. Adapun alur penelitian yang diterapkan dalam penelitian ini disajikan pada Gbr 1.



Gbr. 1 Diagram Alur Penelitian.

A. Mengidentifikasi Masalah

Pada tahapan penelitian ini diawali dengan mengidentifikasi masalah. Penelitian ini berfokus pada permasalahan kondisi kantuk yang dapat menurunkan konsentrasi dan meningkatkan resiko kecelakaan pengemudi. Untuk mengatasi permasalahan tersebut, penelitian ini akan melakukan perancangan model dan pengimplementasian pada dataset yang bertujuan untuk membuat model yang dapat mengklasifikasikan mata tertutup dan mata terbuka. Pada penelitian ini juga menerapkan replikasi channel dengan menggandakan channel citra *grayscale* menjadi tiga channel agar sesuai dengan format input arsitektur *MobileNetV2* sehingga dapat lebih optimal serta meningkatkan akurasi dari sistem deteksi kelelahan berbasis mata.

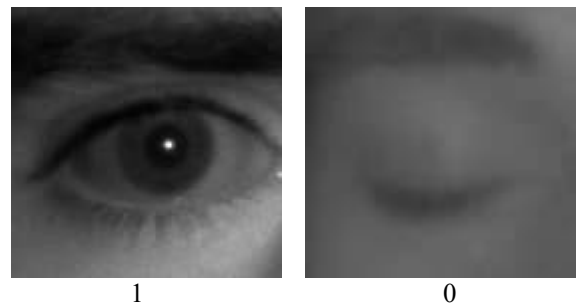
B. Studi Literatur

Pada penelitian ini melakukan studi literatur dengan tujuan mencari sumber literatur yang dapat digunakan sebagai pendukung dan acuan penelitian. Sumber literatur dari berbagai macam sumber seperti buku, artikel, dan jurnal terkait *Deep Transfer Learning* dan Replikasi channel pada citra *grayscale*. studi literatur dilakukan untuk memahami konsep dasar sekaligus meninjau penelitian terdahulu. Tahapan ini dapat membantu peneliti merancang eksperimen yang sesuai serta memperoleh hasil penelitian yang lebih optimal.

C. Deskripsi Data

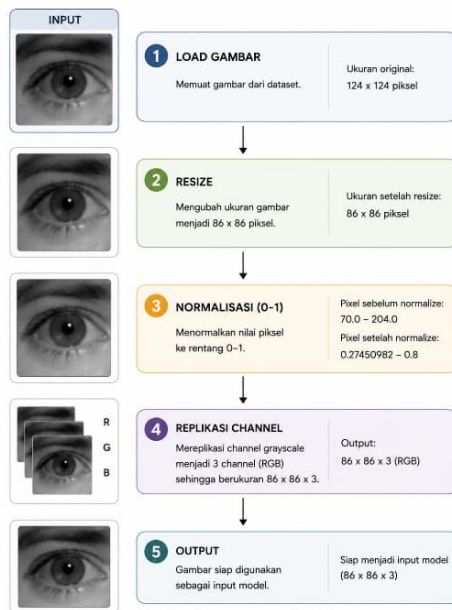
dataset yang digunakan dalam penelitian ini berasal dari dataset milik Prasad V Patil yang tersedia pada platform kaggle. Dataset Deteksi Kelelahan berbasis mata

dibuat menggunakan dataset MRL dan *Closed Eyes in the Wild* (CEW), serta dataset unik milik Prasad V Patil. Dataset berskala besar ini yang terdiri dari citra mata manusia dalam kondisi tertutup dan terbuka dalam format *grayscale* yang dapat digunakan untuk deteksi mata dan selanjutnya dapat dikembangkan untuk deteksi kelelahan. Pada penelitian ini dataset yang digunakan dataset versi 4 berjumlah 4000 citra mata yang terdiri dari dua kategori yaitu 2000 citra mata tertutup dan 2000 citra mata terbuka. pada dataset tersebut terdapat 2 label yaitu 0 untuk mata tertutup dan 1 untuk mata terbuka[10]. Data yang digunakan dalam penelitian ini dibagi ke dalam dua bagian utama, yaitu data latih dan data uji dengan rasio 70% untuk data latih dan 30% untuk data uji. Selanjutnya, sebagian dari data latih tersebut digunakan sebagai data validasi dalam proses pelatihan model. Berikut merupakan beberapa contoh sample data dari MRL Eye Dataset yang digunakan dalam penelitian ini. Contoh citra ini ditampilkan untuk memberikan gambaran awal mengenai bentuk data input, variasi kondisi citra, serta label yang digunakan dalam proses klasifikasi.



Gbr. 2 Sample Data Citra Mata

D. Preprocessing



Gbr. 3 Alur Preprocessing

Preprocessing adalah proses pengolahan data input yang bertujuan mempersiapkan data agar siap digunakan pada tahap selanjutnya, sehingga dapat meminimalkan risiko *overfitting* serta meningkatkan kemampuan generalisasi model [11]. Pada gambar 3 menunjukkan alur *Preprocessing* citra akan melalui beberapa tahapan yaitu *load* gambar, *resize*, replikasi *channel*, dan normalisasi. pertama tahap *load* gambar dimana seluruh citra dari dataset akan dimuat kedalam program. Tujuan dari proses tersebut agar data dapat diakses dan diolah ke tahap berikutnya. Selanjutnya seluruh citra akan diubah ukuran menjadi 86 x 86 pixel.

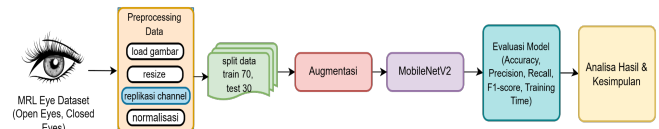
$$I_{RGB} = I_{gray} \otimes [1,1,1] \quad (1)$$

Di sini, I_{RGB} adalah gambar RGB yang telah diubah. I_{gray} adalah gambar mata grayscale asli, dan $[1, 1, 1]$ adalah vektor yang digunakan untuk mereplikasi saluran grayscale [8]. Selain itu, diberikan label kelas pada masing-masing kategori. Selanjutnya, citra dinormalisasi dengan membaginya dengan 255 sehingga berada pada rentang 0,1.

$$x' = \frac{x - \min(x)}{(x) - \min(x)} \quad (2)$$

Menormalkan nilai piksel dari ruang piksel asli (0–255) ke (0,1) [14]. Proses normalisasi tersebut bertujuan mempercepat proses pelatihan dan meningkatkan stabilitas model [2].

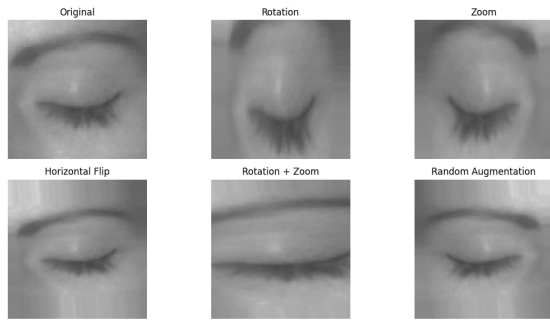
E. Perancangan Model Klasifikasi



Gbr. 4 Alur Perancangan Model

Gambar 4 menunjukkan alur perancangan Model yang akan dirancang dalam penelitian ini menerapkan metode klasifikasi citra berbasis deep learning dengan mengimplementasikan pendekatan deep transfer learning. Penelitian ini menggunakan model *MobileNetV2* untuk menemukan arsitektur yang paling optimal dalam klasifikasi citra mata terbuka dan mata tertutup. Langkah - langkah perancangan model klasifikasi adalah sebagai berikut:

1. Menginput dataset citra berupa *MRL Eye Dataset* yang terdiri dari citra mata terbuka (*open eyes*) dan mata tertutup (*closed eyes*).
2. *Preprocessing* data pada citra dengan mengubah ukuran citra menjadi 86×86 piksel, melakukan normalisasi piksel ke rentang 0,1 , serta replikasi channel agar citra *grayscale* dapat diubah menjadi RGB. Replikasi channel itu bukan metode pengolahan fitur, tapi hanya penyesuaian input karena *MobileNetV2* membutuhkan 3 channel. Proses replikasi channel tidak mengubah informasi visual citra, melainkan hanya menduplikasi nilai piksel ke dalam tiga channel identik untuk menyesuaikan format input model CNN. Dataset pada penelitian ini berupa citra grayscale dengan satu channel. Sementara itu, arsitektur model seperti *MobileNet* memerlukan input dengan tiga channel (RGB). Oleh karena itu, dilakukan proses replikasi channel dengan menggandakan channel grayscale menjadi tiga channel identik sehingga menghasilkan citra berformat (H, W, 3). Proses ini tidak menambahkan informasi baru, melainkan hanya bertujuan untuk memastikan kompatibilitas input dengan arsitektur model. Arsitektur CNN modern seperti *MobileNetV2* umumnya menggunakan citra berformat RGB sebagai input.
3. Split dataset, dimana data dibagi menjadi data latih dan data uji dengan perbandingan tertentu. Sebanyak 70% data *training*(50% data *training*, 20% data *validation*) dan 30% data *testing*.
4. Augmentasi data merupakan proses untuk memperbanyak serta memperkaya variasi dataset tanpa perlu mengumpulkan data baru secara manual. Teknik ini diterapkan untuk meningkatkan generalisasi model dan mencegah *overfitting*, terutama ketika jumlah data latih terbatas.

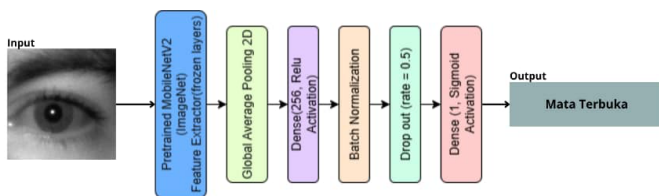


Gbr. 5 Proses Augmentasi data

Gambar 5 menunjukkan proses data augmentation yang diterapkan pada citra mata untuk meningkatkan variasi data pelatihan. Teknik augmentasi yang digunakan meliputi rotasi, zoom, horizontal flip, serta kombinasi transformasi.

Dalam penelitian ini, augmentasi dilakukan dengan menerapkan transformasi seperti rotasi acak, flip, dan pembesaran (zooming) pada citra mata. Langkah ini membantu model mengenali pola citra dengan sudut pandang dan posisi yang bervariasi, sehingga kinerja deteksi kelelahan menjadi lebih stabil dan akurat. Pada penelitian ini, proses augmentasi data dilakukan menggunakan *ImageDataGenerator* dengan parameter rotasi acak sebesar $\pm 15^\circ$, pembesaran (zoom) hingga 20%, dan pembalikan horizontal secara acak. Teknik augmentasi ini bertujuan untuk memperkaya variasi citra mata pada dataset pelatihan, sehingga model dapat beradaptasi terhadap perubahan posisi dan orientasi mata di kondisi nyata serta mengurangi risiko *overfitting*. [12].

5. *Training model*, yaitu data latih digunakan untuk melatih model *deep learning* dengan arsitektur *MobileNetV2*.
6. Evaluasi model dilakukan dengan mengukur performa menggunakan metrik akurasi, presisi, recall, *F1-score*, dan *training time*.
7. Analisis hasil pemilihan model terbaik, dimana model dengan kinerja paling optimal dan efisien untuk proses klasifikasi citra mata pada sistem deteksi mata tertutup dan terbuka.



Gbr. 6 Alur Arsitektur MobileNetV2

Gambar 6 menunjukkan alur Arsitektur *pretrained model MobileNetV2* yang telah dilatih menggunakan dataset *ImageNet*. *MobileNetV2* berperan sebagai *feature extractor* dengan seluruh lapisan konvolusionalnya dibekukan (*frozen layers*), sehingga bobot dari pelatihan sebelumnya tidak

berubah dan dapat digunakan untuk mengekstraksi fitur-fitur penting dari citra input. Arsitektur *MobileNet* menggunakan *Batch Normalization (BN)* dan unit *Rectified-Linear (ReLU)* untuk konvolusi kedalaman dan konvolusi titik.

lapisan *Global Average Pooling 2D*, yang berfungsi untuk mengubah peta fitur berdimensi tinggi menjadi vektor satu dimensi dengan cara merata-ratakan nilai setiap fitur. Lapisan tersebut membantu mengurangi jumlah parameter dan mencegah terjadinya *overfitting*. Lapisan *batch normalization* untuk menormalkan nilai aktivasi agar pelatihan lebih stabil, mempercepat konvergensi, dan membantu mencegah *overfitting*. *Dropout* dengan tingkat 0.5 digunakan untuk menonaktifkan 50% neuron secara acak pada saat pelatihan, yang berfungsi sebagai mekanisme regularisasi untuk meningkatkan kemampuan generalisasi model. lapisan *Dense* berukuran 1 dengan fungsi aktivasi *Sigmoid* untuk menghasilkan output biner, yaitu mengklasifikasikan citra ke dalam dua kategori yaitu mata tertutup atau terbuka [13].

F. Pengukuran Kinerja Model

Pengukuran kinerja model dilakukan untuk mengetahui sejauh mana model klasifikasi mampu mengenali citra mata terbuka dan tertutup dengan benar. Evaluasi efisiensi model yang diusulkan didasarkan pada lima jenis metrik pengukuran kinerja yaitu akurasi, presisi, recall, *F1-score*, dan *training time*. Penggabungan keempat metrik kinerja tersebut memberikan penilaian komprehensif terhadap efektivitas model dalam klasifikasi citra. Meskipun akurasi memberikan gambaran umum tentang kinerja model, presisi, recall, dan *F1-score*. memberikan wawasan tentang kemampuan model untuk mengidentifikasi dengan benar, serta keseimbangannya antara presisi dan recall [9]. Selain menggunakan metrik berbasis hasil klasifikasi. Penelitian ini juga mempertimbangkan *training time* sebagai indikator efisiensi. Pengukuran waktu *training* dilakukan untuk mengetahui seberapa cepat model dapat *mentraining* dataset.

Akurasi yaitu mengukur perbandingan antara jumlah prediksi yang benar dengan jumlah keseluruhan data uji [14].

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

Presisi merupakan rasio antara jumlah contoh positif yang diprediksi dengan benar terhadap jumlah contoh positif yang diprediksi [11]. Hal ini sangat berguna ketika biaya dari *false positives* sangat kritis. Presisi dihitung sebagai berikut:

$$Presisi = \frac{TP}{TP + FP} \quad (4)$$

Recall yaitu kemampuan model dalam menemukan seluruh data positif yang sebenarnya [4]. Efektivitas klasifikasi dihitung menggunakan recall, yang diberikan oleh persamaan berikut:

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

F1 score adalah rata-rata harmonik dari presisi dan recall. Skor ini sangat berguna ketika Anda ingin menyeimbangkan presisi dan recall, dan Anda ingin satu metrik yang mempertimbangkan keduanya. Skor F1 tinggi ketika baik presisi maupun recall tinggi [9].

$$F1\ score = \frac{2 * (Precision * Recall)}{Precision + Recall} \quad (6)$$

Training time yaitu mengukur waktu training untuk menilai efisiensi model. Training time dapat berbeda tergantung pada arsitektur dan kompleksitas model. Menghitung waktu training model *MobileNetV2* dilakukan untuk menilai efisiensi arsitektur tersebut [8].

G. Menganalisa Hasil

Pada penelitian ini menyajikan analisis terperinci dari hasil penelitian yang diperoleh dari konfigurasi model pada berbagai arsitektur dan menyelidiki kinerja model terhadap variasi jumlah *epoch*. Kinerja model dievaluasi menggunakan empat metrik utama yaitu *accuracy*, *precision*, *recall*, dan *f1-score*. Teknik dalam menganalisa hasil dalam penelitian ini yaitu mengamati grafik *accuracy* dan *loss* pada hasil training data maupun validasi untuk mengetahui pola hasil training model, Melakukan evaluasi *confusion matrix* untuk menilai hasil prediksi, Analisis ini bertujuan untuk mengidentifikasi konfigurasi pelatihan yang paling optimal dan menganalisa sejauh mana pengaruh metode replikasi channel mampu meningkatkan performa klasifikasi pada citra mata terbuka dan tertutup untuk deteksi kelelahan[15].

III. HASIL DAN PEMBAHASAN

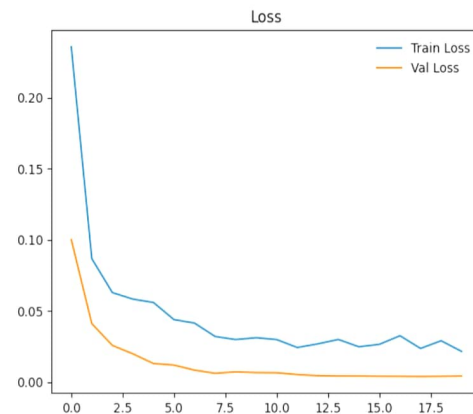
Berdasarkan percobaan yang telah dilakukan, model *MobileNetV2* menunjukkan performa yang baik dalam tugas klasifikasi citra mata untuk deteksi kelelahan. Pengujian dilakukan dengan berbagai kombinasi hyperparameter, meliputi learning rate, batch size, dropout rate, dan jumlah epoch, untuk memperoleh konfigurasi model yang optimal.

TABEL I
Hasil Pelatihan Model *MobileNetV2*

Learning rate	Batch Size	Drop out Rate	Epoch	Train accuracy	Val accuracy	test accuracy	F1 score	Training Time
0,0001	16	0,5	10	94.79%	98.58%	98.42%	0.97	127.02
0,00001	16	0,3	10	96.57%	98.40%	98.09%	0.97	106.12
0,0001	32	0,3	20	99.51%	100%	99.92%	1.00	178.75
0,00001	32	0,3	10	96.17%	99.29%	98.50%	0.98	98.61
0,00001	16	0,5	10	96.17%	99.29%	99.75%	0.99	357.33

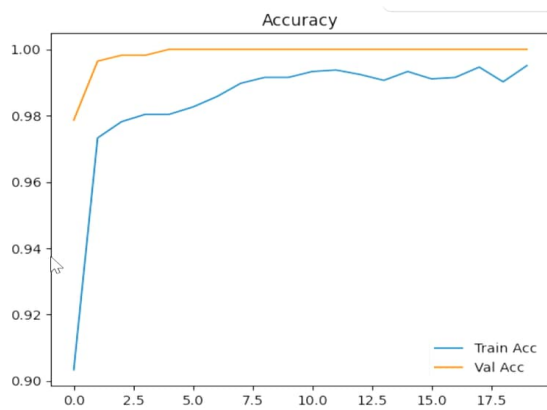
Berdasarkan Tabel I, model *MobileNetV2* menunjukkan performa yang sangat baik dalam klasifikasi citra mata untuk deteksi kelelahan. Seluruh konfigurasi menghasilkan train accuracy di atas 94% serta validation dan test accuracy di atas

98%, yang menunjukkan kemampuan model dalam mengenali pola citra secara optimal. Konfigurasi terbaik diperoleh pada learning rate 0,0001, optimizer Adam, batch size 32, dropout rate 0,3, dan epoch 20 dengan train accuracy sebesar 99,51%, validation accuracy 100%, test accuracy 99,92%, serta *F1-score* 1,00. Hasil tersebut menunjukkan bahwa kombinasi hyperparameter tersebut mampu menghasilkan performa klasifikasi paling optimal dengan tingkat kesalahan yang sangat rendah. Selain itu, penggunaan learning rate 0,0001 dengan batch size 32 dan dropout rate 0,3 terbukti memberikan stabilitas pelatihan yang lebih baik dibandingkan konfigurasi lainnya. Waktu pelatihan yang dibutuhkan pada konfigurasi tersebut sekitar 178,75 detik, yang menunjukkan bahwa model mampu mencapai performa tinggi dengan waktu komputasi yang relatif efisien.



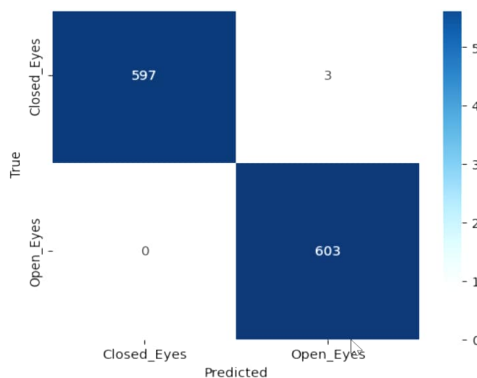
Gbr. 7 Grafik Loss Training dan Validasi Model

Berdasarkan Gambar 7, grafik loss training dan validasi menunjukkan penurunan yang signifikan seiring bertambahnya epoch. Pada awal pelatihan, nilai training loss dan validation loss masih cukup tinggi, namun mengalami penurunan secara bertahap hingga mendekati nol. Hal ini menunjukkan bahwa model *MobileNetV2* mampu mempelajari pola data dengan baik selama proses pelatihan. Selain itu, validation loss memiliki nilai yang lebih rendah dan stabil dibandingkan training loss. Kondisi ini menunjukkan bahwa model memiliki kemampuan generalisasi yang baik dan tidak mengalami *overfitting* secara signifikan. Dengan demikian, hasil grafik *loss* menunjukkan bahwa model telah terlatih secara optimal dan mampu menghasilkan performa klasifikasi yang baik pada data validasi.



Gbr. 8 Grafik Akurasi Training dan Validasi Model

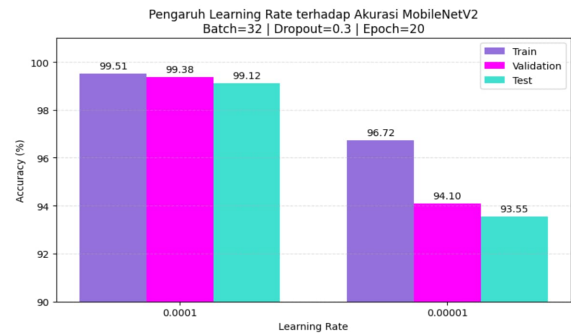
Gambar. 8 menunjukkan bahwa akurasi training dan validasi mengalami peningkatan secara konsisten seiring bertambahnya jumlah epoch. Pada tahap awal pelatihan, nilai training accuracy masih berada di kisaran 90%, kemudian meningkat secara bertahap hingga mencapai lebih dari 99%. Di sisi lain, validation accuracy meningkat dengan cepat dan cenderung stabil pada nilai mendekati 100%. Hal tersebut menunjukkan bahwa model *MobileNetV2* mampu mengenali pola pada data dengan sangat baik sehingga menghasilkan tingkat akurasi klasifikasi yang tinggi. Selain itu, perbedaan nilai antara akurasi training dan validasi yang tidak terlalu besar mengindikasikan bahwa model memiliki kemampuan mengklasifikasi dengan baik dan tidak menunjukkan gejala overfitting yang signifikan selama proses pelatihan.



Gbr. 9 Confusion Matrix dan Classification report Model

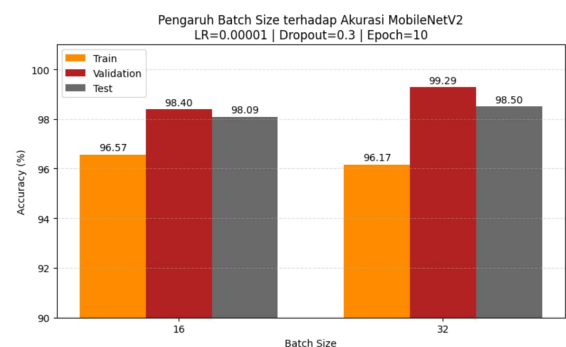
Berdasarkan Gambar. 9, confusion matrix menunjukkan bahwa model *MobileNetV2* mampu melakukan klasifikasi citra mata dengan sangat baik. Pada kelas *Closed_Eyes*, sebanyak 597 data berhasil diprediksi dengan benar, sedangkan 3 data salah diklasifikasikan sebagai *Open_Eyes*. Pada kelas *Open_Eyes*, seluruh 603 data berhasil diprediksi dengan benar tanpa kesalahan klasifikasi. Hasil tersebut menunjukkan bahwa model memiliki tingkat akurasi dan kemampuan klasifikasi yang sangat tinggi dalam membedakan kondisi mata terbuka dan tertutup. Jumlah kesalahan prediksi yang sangat sedikit mengindikasikan bahwa model mampu mengenali fitur citra secara efektif serta memiliki performa yang stabil pada data pengujian.

A. Analisis Hasil Penelitian



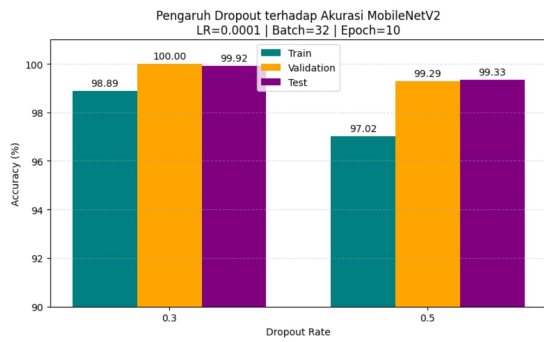
Gbr. 10 Pengaruh Learning Rate terhadap akurasi model

Pada gambar 10 dapat terlihat Pola pengaruh learning rate yang serupa juga terlihat pada model *MobileNetV2*. Penggunaan learning rate 0,0001 menghasilkan proses pelatihan yang lebih cepat konvergen dan stabil, dengan selisih akurasi pelatihan dan validasi yang kecil serta nilai akurasi pengujian yang konsisten tinggi. Pada konfigurasi ini, model menunjukkan kemampuan generalisasi yang baik di berbagai kombinasi batch size dan dropout. Sebaliknya, ketika menggunakan learning rate 0,00001, performa model cenderung lebih fluktuatif dan pada beberapa konfigurasi mengalami penurunan akurasi validasi dan pengujian. Hal ini menunjukkan bahwa meskipun *MobileNetV2* menggunakan bobot pre-trained, nilai learning rate yang terlalu kecil tetap dapat memperlambat penyesuaian bobot akhir terhadap karakteristik dataset penelitian.



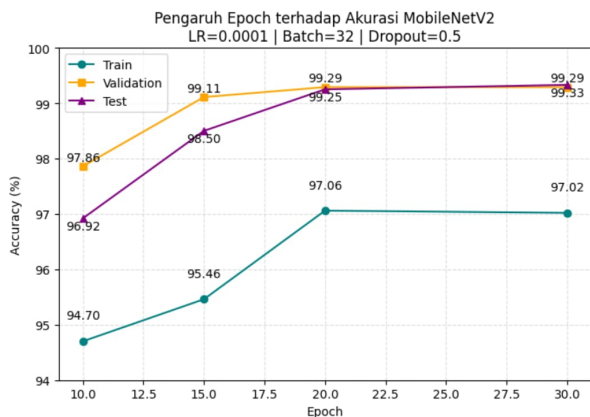
Gbr. 11 Visualisasi Pengaruh Batch Size terhadap akurasi model

Pada model *MobileNetV2*, pengaruh batch size, jumlah epoch, dan dropout rate terhadap performa model juga terlihat cukup jelas. pada gambar 11 menunjukkan bahwa batch size 32 menghasilkan performa yang lebih konsisten dibandingkan batch size 16, terutama pada kestabilan akurasi validasi dan pengujian. Hal ini menunjukkan bahwa ukuran batch yang lebih besar membantu proses pembaruan bobot menjadi lebih stabil pada arsitektur berbasis pre-trained.



Gbr. 12 Visualisasi Pengaruh Dropout Rate terhadap akurasi model

Terkait dropout gambar 12 menjelaskan bahwa nilai 0,3 memberikan hasil paling seimbang antara kemampuan generalisasi dan kestabilan pelatihan. Ketika dropout ditingkatkan menjadi 0,5, performa tidak meningkat secara signifikan dan pada beberapa konfigurasi justru sedikit menurun. Hal ini menunjukkan bahwa MobileNetV2 tidak memerlukan regularisasi dropout yang terlalu tinggi karena arsitekturnya sudah cukup efisien dalam mengekstraksi fitur.



Gbr. 13 Visualisasi Pengaruh Epoch terhadap akurasi model

Dari sisi *epoch* yg terlihat pada gambar 13, peningkatan jumlah *epoch* hingga sekitar 15–20 memberikan kenaikan akurasi yang nyata. Setelah melewati kisaran tersebut, peningkatan performa cenderung kecil dan mulai menunjukkan kondisi konvergen, sehingga penambahan epoch lebih lanjut tidak selalu memberikan keuntungan signifikan.

IV. KESIMPULAN

Berdasarkan hasil penelitian, model *MobileNetV2* menunjukkan kinerja yang sangat baik dalam klasifikasi citra mata untuk deteksi kelelahan. Konfigurasi optimal diperoleh pada *learning rate* 0,0001, batch size 32, dropout rate 0,3, dan 20 *epoch*, dengan menghasilkan akurasi data uji sebesar 99,92% dan *F1-score* sebesar 1,00. Hasil ini menunjukkan bahwa model memiliki kemampuan klasifikasi yang tinggi serta mampu melakukan generalisasi dengan baik. Analisis hyperparameter menunjukkan bahwa *learning rate* 0,0001

memberikan konvergensi yang lebih cepat dan stabil dibandingkan nilai yang lebih kecil. Penggunaan batch size 32 meningkatkan kestabilan pelatihan, sementara jumlah epoch pada kisaran 15–20 memberikan performa optimal sebelum model mencapai kondisi konvergen. Selain itu, dropout rate sebesar 0,3 terbukti memberikan keseimbangan terbaik antara kemampuan generalisasi dan kestabilan model. Secara keseluruhan, *MobileNetV2* sebagai model berbasis *pre-trained convolutional neural network* mampu memberikan performa yang akurat dan efisien dalam deteksi kelelahan berbasis citra mata, sehingga berpotensi untuk diimplementasikan pada sistem pemantauan kondisi pengemudi atau pengguna secara real-time.

UCAPAN TERIMAKASIH

Puji syukur kehadirat Tuhan Yang Maha Esa atas rahmat-Nya sehingga jurnal ini dapat diselesaikan. Penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam proses penyusunan jurnal ini. terutama kepada dosen pembimbing atas kesabaran, bimbingan, serta ilmu yang diberikan. terima kasih juga kepada keluarga yang selalu memberikan doa dan dukungan, serta teman-teman yang turut membantu dan memberikan semangat selama proses penelitian.

REFERENSI

- [1] R. Florez, F. Palomino-Quispe, A. B. Alvarez, R. J. Coaquira-Castillo, and J. C. Herrera-Levano, "A Real-Time Embedded System for Driver Drowsiness Detection Based on Visual Analysis of the Eyes and Mouth Using Convolutional Neural Network and Mouth Aspect Ratio," *Sensors*, vol. 24, no. 19, Oct. 2024, doi: 10.3390/s24196261.
- [2] H. Wang, "Driver fatigue detection based on lightweight MobileNetV2," *Applied and Computational Engineering*, vol. 6, no. 1, pp. 1148–1153, Jun. 2023, doi: 10.54254/2755-2721/6/20230513.
- [3] N. Jamil, M. H. Mohd Fadhill, R. Hamzah, and M. I. Ramli, "Comparing MobileNet-SSD and YOLO v3 Learning Architecture for Real-time Driver's Fatigue Detection," *International Journal of Academic Research in Business and Social Sciences*, vol. 11, no. 12, Dec. 2021, doi: 10.6007/ijarbs/v11-i12/11984.
- [4] M. Venkateswarlu and V. Rami Reddy Ch, "DrowsyDetectNet: Driver Drowsiness Detection Using Lightweight CNN With Limited Training Data," *IEEE Access*, vol. 12, pp. 110476–110491, 2024, doi: 10.1109/ACCESS.2024.3440585.
- [5] M. Goma, R. Mahmoud, and A. Sarhan, "A CNN-LSTM-based Deep Learning Approach for Driver Drowsiness Prediction," *Journal of Engineering Research*, vol. 0, no. 0, pp. 0–0, Aug. 2022, doi: 10.21608/erjeng.2022.141514.1067.
- [6] A. Bhanja, D. Parhi, D. Gajendra, K. Sinha, and A. K. Sahoo, "Driver drowsiness shield (DDSH): a real-time driver drowsiness detection system," *ROBOMECH Journal*, vol. 12, no. 1, Dec. 2025, doi: 10.1186/s40648-025-00307-4.
- [7] S. Lakshmi and A. Vinoth, "CNN-Powered Driver Fatigue Detection: Evaluating InceptionV3, VGG19, and ResNet50V2," *Communications on Applied Nonlinear Analysis* Vol 32 No. 9s. 2025. doi: https://doi.org/10.52783/cana.v32.4520.
- [8] C. Gu and M. Lee, "Deep Transfer Learning Using Real-World Image Features for Medical Image Classification, with a Case Study on Pneumonia X-ray Images," *Bioengineering*, vol. 11, no. 4, Apr. 2024, doi: 10.3390/bioengineering11040406.
- [9] D. Salem and M. Waleed, "Drowsiness detection in real-time via convolutional neural networks and transfer learning," *Journal of Engineering and Applied Science*, vol. 71, no. 1, Dec. 2024, doi: 10.1186/s44147-024-00457-z.

- [10] Prasad V Patil, "MRL Eye Dataset," Kaggle. [Online]. Available: <https://www.kaggle.com/datasets/prasadvpatil/mrl-dataset/data>[Diakses: 12-Aug-2025]..
- [11] O. F. Hassan, A. F. Ibrahim, A. Gomaa, M. A. Makhlof, and B. Hafiz, "Real-time driver drowsiness detection using transformer architectures: a novel deep learning approach," *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-02111-x.
- [12] I. Jahan et al., "4D: A Real-Time Driver Drowsiness Detector Using Deep Learning," *Electronics (Switzerland)*, vol. 12, no. 1, Jan. 2023, doi: 10.3390/electronics12010235
- [13] H. Hartatik and M. K. Anam, "Comparison of Convolutional Neural Network Architecture on Detection of Helmet Use by Humans," *Elinvo (Electronics, Informatics, and Vocational Education)*, vol. 8, no. 1, pp. 44–54, Jun. 2023, doi: 10.21831/elinvo.v8i1.52104.
- [14] R. Florez, F. Palomino-Quispe, R. J. Coaquira-Castillo, J. C. Herrera-Levano, T. Paixão, and A. B. Alvarez, "A CNN-Based Approach for Driver Drowsiness Detection by Real-Time Eye State Identification," *Applied Sciences (Switzerland)*, vol. 13, no. 13, Jul. 2023, doi: 10.3390/app13137849.
- [15] S. H. Al-Gburi et al., "EffRes-DrowsyNet: A Novel Hybrid Deep Learning Model Combining EfficientNetB0 and ResNet50 for Driver Drowsiness Detection," *Sensors*, vol. 25, no. 12, Jun. 2025, doi: 10.3390/s25123711.