

BAB III

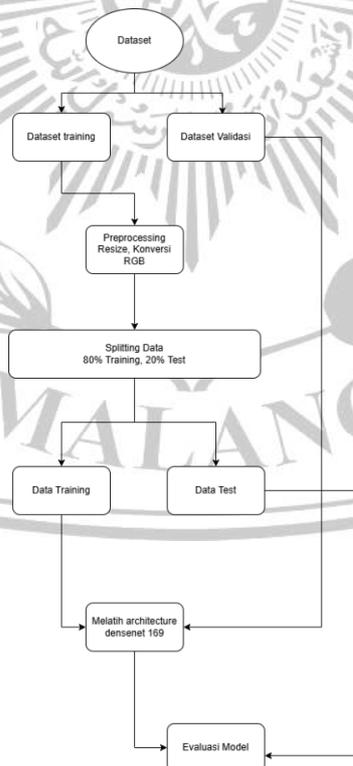
METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Penelitian ini menggunakan metode kuantitatif. Meskipun data yang digunakan berupa citra, dalam penelitian ini data tersebut diolah menjadi bentuk numerik melalui proses ekstraksi fitur oleh model *DenseNet-169*. Hasilnya berupa angka yang merepresentasikan pola dan karakteristik dari citra tersebut. Data numerik ini kemudian dianalisis menggunakan metode statistik untuk mengevaluasi performa model. Penelitian kuantitatif bertujuan untuk memperoleh hasil yang objektif dan terukur dengan menggunakan metrik seperti *akurasi*, *precision*, *recall*, dan *F1-score*. Hasil analisis ini disajikan dalam bentuk tabel, grafik, dan *confusion matrix* untuk mendukung interpretasi hasil dan kesimpulan penelitian.

3.2 Tahapan Penelitian

Penelitian ini dilakukan dengan menjaankan beberapa tahap yaitu Download data, ekstraksi data, *Preprocessing*, *splitting data*, melatih model dan evaluasi. Tahapan penelitian secara jelas dapat dilihat pada Gambar 3.1.



Gambar 3.1. Rancangan Penelitian

3.3 Dataset

Pada penelitian ini, dataset yang digunakan adalah dataset deteksi pengemudi terdistraksi dari State Farm yang tersedia di Kaggle. Dataset ini khusus dirancang untuk mengidentifikasi berbagai tingkat distraksi pada pengemudi yang sedang mengemudi. Ketika di download terdapat dua folder yaitu folder data training dan data validasi. Untuk data training terdapat 10 klasifikasi tingkah laku yang menjadi fokus dalam penelitian ini, yang masing-masing merepresentasikan tingkat distraksi berbeda. *Normal Driving, Texting – Right, Talking on the Phone – Right, Texting – left, Talking on the Phone – left, Operating the Radio, Drinking, Reaching Behind, Touching Face* dan *Talking to Passenger*. Berikut contoh gambar untuk kelasnya masing masing

Tabel 3. 2 Gambar di setiap kelas

NO	Kelas	Gambar
1	<i>Normal Driving</i>	
2	<i>Texting – Right</i>	
3	<i>, Talking on the Phone – Right</i>	
4	<i>Texting – left</i>	

5	<i>, Talking on the Phone – left</i>	
6	<i>Operating the Radio</i>	
7	<i>Drinking</i>	
8	<i>Reaching Behind</i>	
9	<i>, Touching Face</i>	
10	<i>Talking to Passenger</i>	

Setiap kelas dalam dataset ini memiliki jumlah citra yang sama, yaitu 1.900 citra per kelas, sehingga total keseluruhan adalah 19.000 *citra*. Distribusi yang seragam ini membantu dalam mengurangi bias selama proses pelatihan model pembelajaran mesin. Untuk data validasi digunakan untuk validasi

3.4 Pembacaan Data

Data Data citra yang digunakan dalam penelitian ini dibaca dan diolah menggunakan fungsi *get_cv2_image()*, yang memanfaatkan pustaka *OpenCV* (*cv2*). Proses ini bertujuan untuk memastikan citra diubah menjadi format dan ukuran yang sesuai sebelum dimasukkan ke dalam model. *OpenCV* adalah pustaka yang dirancang untuk memberikan akses ke algoritma pemrosesan citra yang efisien dan mudah digunakan. Fungsi ini memungkinkan proses konversi citra menjadi format yang dapat dianalisis lebih lanjut, termasuk teknik *resizing*. *OpenCV* mendukung berbagai aplikasi visi komputer, memungkinkan pengguna untuk melakukan tugas-tugas kompleks dengan sederhana dan efisien [8].

3.5 Preprocessing

Preprocessing dalam penelitian ini melibatkan serangkaian langkah penting untuk mempersiapkan data citra agar siap digunakan oleh model pembelajaran mesin. Data *preprocessing* yang dilakukan dalam penelitian ini adalah sebagai berikut:

3.5.1 Resize

Proses mengubah ukuran gambar atau foto. Ini melibatkan perubahan resolusi atau dimensi horizontal dan vertikal dari gambar tersebut. *Resize* dapat dilakukan untuk memperkecil atau memperbesar gambar sesuai dengan kebutuhan. Penelitian ini menggunakan *resize* sebesar 224 x 224.

3.5.2 Konversi RGB

Mengubah format warna gambar menjadi model warna **RGB** (Red, Green, Blue).

3.6 Splitting Data (Pembagian Data)

Dataset dibagi menjadi dua set: latih dan uji. Dengan proporsi 80% untuk latih, 20% test. Pembagian ini penting untuk mengevaluasi performa model secara objektif. Set latih digunakan untuk melatih model. Sedangkan test untuk menguji model yang sudah di latih

3.7 Densenet 169

DenseNet (Densely Connected Convolutional Networks) adalah arsitektur jaringan saraf konvolusional (*CNN*) yang dirancang untuk mengoptimalkan aliran informasi dan gradien melalui jaringan. Arsitektur ini dikembangkan untuk menyempurnakan efisiensi dan efektivitas dalam proses pembelajaran dengan memanfaatkan konektivitas yang sangat padat antar lapisan. *DenseNet* mencapai efisiensi ini melalui konsep blok *Dense*, di mana setiap lapisan terhubung langsung dengan setiap lapisan lain dalam blok yang sama. Keterhubungan ini memungkinkan setiap lapisan menerima tambahan input dari semua lapisan sebelumnya, menghasilkan koneksi yang dikenal sebagai "*concatenate*" (bukan penjumlahan), yang mendukung '*feature reuse*'—penggunaan ulang fitur secara efektif untuk mengurangi jumlah parameter dan memastikan aliran gradien yang efisien saat pelatihan (9).

DenseNet-169 adalah jaringan saraf konvolusional yang padat, di mana setiap lapisan terhubung dengan semua lapisan berikutnya, menghasilkan $\frac{N(N+1)}{2}$ koneksi untuk N lapisan. Arsitektur ini terdiri dari lapisan konvolusional awal, empat blok padat (*dense block*) dengan lapisan transisi di antara setiap blok, dan diakhiri dengan lapisan klasifikasi yang menggunakan fungsi aktivasi *softmax*. Setiap blok padat terdiri dari konvolusi 1×1 yang diikuti oleh konvolusi 3×3 . Terdapat 6, 12, 32, dan 32 set konvolusi, menghasilkan total 82 set konvolusi. Dengan 164 lapisan konvolusional, tiga lapisan transisi, satu lapisan konvolusional, dan satu lapisan klasifikasi, total keseluruhan adalah 169 lapisan. Model ini memiliki laju pertumbuhan sebesar 32, yang menentukan jumlah fitur baru yang ditambahkan pada setiap blok padat (10).

3.7.1 Kenapa Menggunakan Densenet 169 di data ini

Pemilihan *DenseNet-169* didasarkan pada kompleksitas tugas klasifikasi citra yang memerlukan pengenalan pola visual dari pengemudi yang terdistraksi dalam berbagai posisi dan aktivitas. Dataset *Kaggle State Farm Distracted Driver Detection* terdiri dari 10 kelas, seperti *Normal Driving*, *Texting (kanan/kiri)*, *Talking on the Phone (kanan/kiri)*, *Operating the Radio*, *Drinking*, *Reaching Behind*, *Touching Face*, dan *Talking to Passenger*. Model harus mampu mengenali variasi ini dengan baik.

DenseNet-169 dipilih karena arsitekturnya yang dalam (169 lapisan) dan kemampuannya dalam *feature reuse*, memungkinkan model mempelajari pola visual yang beragam secara efisien. Arsitektur ini juga membantu mengatasi masalah *vanishing gradient*, sehingga pembelajaran menjadi lebih stabil meskipun model terdiri dari banyak lapisan.

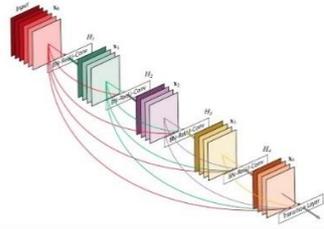
Penelitian oleh Putri (11) menunjukkan bahwa *DenseNet-169* unggul dalam tugas deteksi wajah dengan akurasi 96%. Selain itu, studi oleh Huan (12) membandingkan *DenseNet-169* dengan 12 model lainnya dan mencatat akurasi 66,79% setelah diadaptasi dari ImageNet.

Keunggulan lain *DenseNet-169* adalah kemampuannya menangani dataset kompleks seperti ini tanpa memerlukan banyak parameter tambahan, sehingga beban komputasi tetap efisien.

3.7.2 Struktur DenseNet

DenseNet terdiri dari beberapa blok *Dense* yang dihubungkan melalui lapisan transisi. Setiap lapisan dalam blok *Dense* adalah penerima tambahan dari *output* semua lapisan sebelumnya, membuat struktur menjadi padat. Ini berarti lapisan ke- n dalam sebuah blok memiliki n *input*, yaitu *output* dari semua lapisan 1 hingga $n-1$.

Komponen Utama



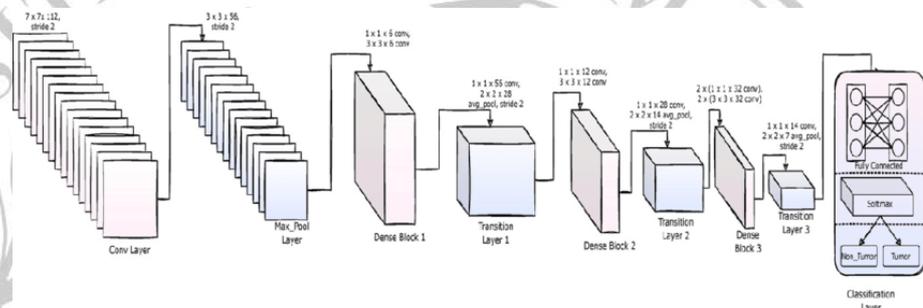
Gambar 3.2

- a) **Blok Dense:** Setiap blok *Dense* terdiri dari beberapa lapisan konvolusi yang masing-masing menerima input dari semua lapisan sebelumnya dalam blok tersebut.
- b) **Lapisan Transisi:** Digunakan antara dua blok *Dense* untuk mengurangi dimensi ruang fitur. Biasanya, ini melibatkan konvolusi 1×1 diikuti oleh pooling rata-rata 2×2 .
- c) **Classification Layer :** Setelah *dense block* terakhir, dilakukan *average pooling* diikuti oleh *fully connected layer* untuk klasifikasi.
- d) **Growth Rate :** Parameter penting dalam *DenseNet* adalah '*growth rate*' yang mengontrol berapa banyak fitur baru yang dihasilkan oleh setiap lapisan dalam sebuah blok *Dense*. Ini mempengaruhi jumlah total parameter dan kompleksitas model
- e) **Width Multiplier (α):** *Hyperparameter* ini mengontrol jumlah filter dalam setiap lapisan. Mengurangi *width multiplier* mengurangi jumlah *parameter* secara proporsional, yang mengurangi kompleksitas model dan kebutuhan komputasi.

3.7.3 Arsitektur Densenet

Model yang digunakan dalam penelitian ini adalah *DenseNet-169*, sebuah arsitektur deep learning yang terkenal karena kemampuannya dalam melakukan klasifikasi gambar dengan efisien. *DenseNet*, atau *Densely Connected Convolutional Networks*, memiliki keunggulan di mana setiap layer terhubung langsung ke setiap layer yang lebih dalam melalui dense connections. Pendekatan ini meningkatkan efisiensi penggunaan fitur serta mempermudah aliran *gradien* selama pelatihan.

Berikut Gambar arsitektur *Densenet-169*



Gambar 3.3 Gambar Arsitektur Densenet 169

Model ini dilatih menggunakan bobot *pre-trained* dari dataset *ImageNet*, yang kemudian digunakan kembali (*transfer learning*) untuk mengekstraksi fitur dari citra yang digunakan dalam penelitian..

Adapun arsitektur model yang digunakan dalam penelitian ini adalah sebagai berikut:

- **Base Model:** *Base model* dari *DenseNet-169* digunakan dengan parameter *include_top=False*, artinya bagian klasifikasi dari model ini tidak digunakan, dan hanya bagian ekstraksi fitur yang dimanfaatkan. Model ini dilatih ulang menggunakan data latih yang ada, tetapi lapisan-lapisan *DenseNet* tidak di-train ulang pada tahap awal untuk menjaga fitur-fitur yang telah dipelajari dari dataset *ImageNet*.

- **Global Average Pooling:** Setelah fitur diekstraksi oleh

DenseNet, digunakan lapisan *GlobalAveragePooling2D* yang mengubah fitur-fitur dua dimensi menjadi satu dimensi dengan menghitung rata-rata dari setiap fitur.

- **Fully Connected Layers:** Setelah lapisan *pooling*, terdapat dua lapisan *Dense* yang masing-masing terdiri dari 512 *neuron* dan 256 *neuron*. Lapisan ini bertanggung jawab untuk melakukan klasifikasi dengan menggunakan fitur yang diekstraksi. Aktivasi yang digunakan pada kedua lapisan ini adalah *ReLU (Rectified Linear Unit)*, yang berfungsi untuk memperkenalkan non-linearitas ke dalam jaringan.

- **Dropout:** Untuk mencegah *overfitting*, digunakan lapisan dropout setelah setiap lapisan *Dense*. Lapisan ini secara acak menonaktifkan sejumlah *neuron* selama proses pelatihan, dengan rasio dropout sebesar 0.5 pada lapisan pertama dan 0.3 pada lapisan kedua.

- **Output Layer:** Lapisan terakhir adalah lapisan *Dense* yang memiliki 10 *neuron*, sesuai dengan jumlah kelas pada data citra (10 kelas pengemudi terdistraksi). Aktivasi yang digunakan adalah *softmax*, yang menghasilkan probabilitas untuk setiap kelas.

3.7.4 Efektivitas Densenet 169

DenseNet-169 memanfaatkan fitur *reuse* untuk membutuhkan lebih sedikit parameter dibandingkan dengan jaringan tradisional lainnya yang memiliki jumlah lapisan dan kedalaman yang sama. Hal ini membuat *DenseNet* lebih efisien dan ringan. Koneksi langsung antar lapisan memfasilitasi aliran gradien selama pelatihan, yang meningkatkan konvergensi dan mengurangi masalah menghilangnya *gradien*. Setiap lapisan menerima input dari semua lapisan sebelumnya, memungkinkan peningkatan integrasi fitur melalui jaringan, yang sangat membantu dalam pembelajaran fitur yang lebih kompleks (13)

3.7.5 DenseNet-169 pada Data Berjumlah Besar dengan Ukuran Resize yang Sama

Dalam Dalam penelitian yang dilakukan oleh Komol (14), berjudul "*Deep Transfer Learning Based Intersection Trajectory Movement Classification for Big Connected Vehicle Data*," model *DenseNet-169* diimplementasikan untuk melakukan klasifikasi arah pergerakan kendaraan menggunakan 19.202 gambar. Hasil penelitian ini menunjukkan bahwa *DenseNet-169* mampu mengelola data dengan sangat baik, mencapai akurasi hingga 99,73% pada gambar berwarna. Selain itu, dibandingkan dengan arsitektur lain seperti VGG19, *DenseNet-169* menunjukkan performa yang lebih unggul dalam mendeteksi gambar berwarna. Penelitian tersebut menggunakan citra berukuran 224x224 dengan *optimizer* "Adam," *learning rate* sebesar 0,0001, dan melibatkan 15 *epoch*.

Pada Pada penelitian lain dengan jumlah data yang sama dan menggunakan *DenseNet-169*, yaitu penelitian berjudul "*Classification of Deepfake Videos Using Pre-trained Convolutional Neural Networks*" oleh Masood (15), model *DenseNet-169* juga diimplementasikan untuk tugas klasifikasi. Penelitian ini membandingkan 10 model deep learning, salah satunya adalah *DenseNet-169*, dengan gambar yang diubah ukurannya menjadi 224x224 dan menggunakan 19.000 data asli. Hasilnya menunjukkan bahwa *DenseNet-169* mencapai akurasi tertinggi sebesar 97,2%, sedangkan model dengan performa terendah adalah VGG-16 dengan akurasi sebesar 89%.

Penggunaan dataset yang besar dalam kedua penelitian ini membuktikan bahwa *DenseNet-169* memiliki kemampuan untuk menangani volume data yang signifikan, serta mampu mengoptimalkan proses pelatihan dengan memanfaatkan kembali fitur-fitur sebelumnya di setiap lapisan (*feature reuse*). Karakteristik *DenseNet* yang dapat mendistribusikan gradien secara lebih efektif

membuat model ini lebih stabil dan efisien, bahkan dalam lingkungan dengan data yang besar. Hal ini menjadi salah satu alasan utama mengapa *DenseNet-169* dipilih dalam penelitian ini untuk mendeteksi pengemudi yang terdistraksi, di mana diperlukan dataset yang besar untuk melatih model dengan berbagai skenario gambar.

3.7.6 Alasan Menggunakan Densenet

Dalam penelitian yang dilakukan oleh Vulli et al. (8) dengan judul "*Fine-Tuned DenseNet-169 for Breast Cancer Metastasis Prediction Using FastAI and 1-Cycle Policy*," *DenseNet-169* diperkenalkan sebagai salah satu arsitektur dalam keluarga *DenseNet* yang memiliki 169 lapisan. Arsitektur ini banyak digunakan dalam tugas klasifikasi berbasis deep learning karena memiliki jumlah parameter yang lebih sedikit dibandingkan dengan varian *DenseNet* lain yang memiliki lapisan lebih sedikit.

Untuk *Face Detection*, *DenseNet-169* juga menunjukkan performa yang baik. Penelitian oleh Nandy (2) menunjukkan bahwa *DenseNet-169* dapat digunakan untuk *face recognition* dengan akurasi tinggi.

Kelebihan utama dari *DenseNet-169* dan varian *DenseNet* lainnya adalah kemampuannya untuk mengatasi masalah vanishing gradient melalui propagasi fitur yang kuat, meminimasi jumlah parameter yang dapat dilatih, serta pemanfaatan kembali fitur (*feature reuse*). Karakteristik ini menjadikan *DenseNet-169* sebagai salah satu arsitektur deep learning yang sangat andal.

Arsitektur *DenseNet-169* melibatkan lapisan-lapisan konvolusi, lapisan pooling maksimum (max-pool), lapisan dense (fully connected), dan lapisan transisi. Model ini menggunakan fungsi aktivasi *ReLU* di seluruh arsitekturnya, sementara lapisan terakhir menggunakan aktivasi *SoftMax*. Lapisan konvolusi bertugas mengekstraksi fitur dari gambar, dan lapisan pooling maksimum

berfungsi untuk mengurangi dimensi input. Setelah itu, lapisan *fully connected* mengikuti lapisan *flatten*, yang berfungsi sebagai jaringan saraf buatan dengan input dalam bentuk *array* tunggal.

Dengan Dengan desain ini, *DenseNet-169* menawarkan keandalan dan efisiensi yang lebih baik dalam tugas klasifikasi gambar, terutama ketika dibandingkan dengan model lain yang membutuhkan lebih banyak parameter atau lapisan yang lebih dalam.

3.8. Pelatihan, Parameter model

Pelatihan model dilakukan dengan menggunakan beberapa parameter penting untuk mengoptimalkan hasil klasifikasi. Berikut penjelasan masing-masing parameter:

1. **BatchSize**

Dalam penelitian ini, digunakan ukuran batch sebesar 50. *Batch size* adalah jumlah sampel data yang diproses sebelum model memperbarui bobot selama pelatihan. Ukuran batch yang lebih besar dapat menggantikan penurunan laju pembelajaran dan memberikan hasil yang sebanding pada kurva pembelajaran, baik pada data pelatihan maupun pengujian. Selain itu, ukuran batch yang besar dapat mengurangi jumlah pembaruan parameter yang diperlukan, sehingga meningkatkan paralelisme dan mengurangi waktu pelatihan (16).

2. **Epoch**

Pelatihan model dilakukan dengan 50 *epoch*. *Epoch* merujuk pada satu siklus penuh di mana algoritma *machine learning* memproses seluruh data pelatihan. Proses ini dilakukan secara iteratif untuk memastikan konvergensi nilai bobot. Pemilihan jumlah epoch yang tepat tidak bisa ditentukan secara langsung, sehingga dilakukan pengujian beberapa nilai *epoch* untuk mencapai akurasi optimal (17).

3. **LearningRate**

Learning rate yang digunakan dalam penelitian ini sebesar 0.0001. *Learning rate* merupakan konstanta proporsionalitas yang menentukan ukuran langkah pembaruan parameter model berdasarkan estimasi gradien. Parameter ini

sangat penting dalam proses optimisasi karena memengaruhi kecepatan dan stabilitas konvergensi fungsi kerugian (18).

3.9 Evaluasi

Evaluasi yang digunakan ialah confusion matrix dan f1 score, precision, recall serta support. *Confusion Matrix* adalah alat untuk analisis prediktif dalam pembelajaran mesin. Ini digunakan untuk memeriksa kinerja model pembelajaran mesin berbasis klasifikasi (19). Dalam konteks klasifikasi multiclass adalah sebuah tabel yang digunakan untuk menggambarkan kinerja model klasifikasi dengan menunjukkan jumlah prediksi yang benar dan salah untuk setiap kelas. Dalam klasifikasi *multiclass*, *confusion matrix* memiliki dimensi $Q \times Q$, di mana Q adalah jumlah kelas yang ada (20).

