

DETEKSI DAN KLASIFIKASI KENDARAAN RINGAN DAN BERAT UNTUK JALAN TOL MENGGUNAKAN CNN

Hilman Fatoni, Diva Ahmad Maulana, Riza Ibnu Adam

Teknik Informatika, Universitas Singaperbangsa Karawang

Jl. HS.Ronggo Waluyo, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat 41361, Telp. (0267) 641177

hilmanfatoni8@gmail.com

ABSTRAK

Dengan meningkatnya volume kendaraan di jalan raya, manajemen lalu lintas yang efisien dan aman menjadi tantangan besar. Deteksi dan klasifikasi jenis kendaraan merupakan komponen penting dalam sistem manajemen jalan tol untuk mendukung pengurangan kemacetan dan peningkatan keamanan. Penelitian ini mengembangkan sistem deteksi dan klasifikasi kendaraan ringan dan berat menggunakan Convolutional Neural Network (CNN) sebagai solusi untuk tantangan tersebut. Dataset terdiri dari 500 gambar kendaraan ringan dan 600 gambar kendaraan berat yang diperoleh melalui web scraping. Proses preprocessing melibatkan resizing, normalisasi, serta pembagian dataset menjadi training dan testing set dengan rasio 80:20. Arsitektur CNN mencakup empat lapisan konvolusi dengan ReLU, lapisan pooling, dan lapisan dense dengan softmax untuk klasifikasi dua kelas. Model dilatih selama 10 epoch dengan batch size 32. Hasil menunjukkan akurasi training 100%, akurasi validasi 100%, dan akurasi testing 100%. Analisis menggunakan classification report dan confusion matrix menunjukkan precision, recall, dan f1-score sebesar 1.00 untuk kedua kelas. Hasil ini mengkonfirmasi potensi CNN dalam meningkatkan kinerja dan keandalan sistem manajemen jalan tol, serta memberikan kontribusi signifikan dalam pengembangan transportasi cerdas.

Kata kunci : Deteksi Kendaraan, Klasifikasi Kendaraan, Convolutional Neural Network, Manajemen Jalan Tol.

1. PENDAHULUAN

Dengan semakin meningkatnya volume kendaraan di jalan raya, manajemen lalu lintas yang efisien dan aman menjadi tantangan besar [1]. Sistem manajemen jalan tol yang efektif tidak hanya membantu dalam mengurangi kemacetan tetapi juga meningkatkan keamanan bagi pengguna jalan. Salah satu komponen kunci dalam sistem ini adalah kemampuan untuk mendeteksi dan mengklasifikasikan jenis kendaraan yang melintas, yang dapat digunakan untuk berbagai tujuan seperti penetapan tarif tol yang berbeda, pengawasan lalu lintas, dan analisis pola lalu lintas.

Kemajuan dalam bidang kecerdasan buatan (AI) dan pembelajaran mesin (ML) telah membuka peluang baru untuk pengembangan sistem deteksi dan klasifikasi kendaraan yang lebih canggih [2]. Convolutional Neural Network (CNN) adalah salah satu teknik pembelajaran mesin yang sangat efektif dalam pemrosesan gambar dan telah menunjukkan hasil yang mengesankan dalam berbagai aplikasi, termasuk pengenalan objek, deteksi wajah, dan analisis citra medis [3].

Penelitian sebelumnya telah menunjukkan berbagai pendekatan dalam deteksi dan klasifikasi. Misalnya, Febri Hafifah dkk. (2021) mengembangkan metode klasifikasi jenis kendaraan menggunakan CNN dengan arsitektur GoogLeNet dan ShuffleNet, menghasilkan akurasi masing-masing sebesar 95,88% dan 96,48% [5]. Didit Iswantoro dan Dewi Handayani UN (2022) menerapkan CNN untuk klasifikasi penyakit tanaman jagung, mencapai akurasi training 97,5%, akurasi validasi 100%, dan akurasi testing 94% [6]. Fani Nurona Cahya dkk. (2021) menggunakan CNN untuk klasifikasi penyakit mata dengan

arsitektur AlexNet, mencapai akurasi 98,37% [7]. Ryas Rafi Karim dan Angga Herlangga (2023) mengimplementasikan CNN dengan metode transfer learning untuk klasifikasi senjata tradisional Jawa Barat, mencapai akurasi 98,44% pada evaluasi model meskipun ada kekurangan pada kelas gacok [8]. Terakhir, Muhammad Eko Prasetyo dkk. (2023) melakukan penelitian tentang klasifikasi ragam kendaraan menggunakan CNN, menunjukkan akurasi 87,12% dengan model yang terdiri dari tiga lapisan konvolusi [9]. Hal ini menunjukkan bahwa penggunaan CNN dalam berbagai bidang seperti pengawasan lalu lintas, pertanian, kesehatan, dan pengenalan objek memberikan hasil yang menjanjikan dalam hal akurasi dan efektivitas. Hal ini menegaskan potensi besar CNN dalam meningkatkan berbagai sistem dan aplikasi yang bergantung pada analisis citra digital.

Dalam rangka untuk memperluas pemahaman dan memperkaya aplikasi dalam bidang pemrosesan citra digital, studi ini berfokus pada pemanfaatan Convolutional Neural Network (CNN) dalam mendeteksi dan mengklasifikasikan kendaraan dengan akurasi dan efisiensi yang tinggi. Penelitian ini bertujuan untuk mengembangkan sistem yang dapat mendeteksi dan mengklasifikasikan kendaraan ringan dan berat secara akurat menggunakan CNN. Data gambar kendaraan diperoleh melalui teknik web scraping dari berbagai sumber online, yang kemudian diproses dan digunakan untuk melatih model CNN. Proses preprocessing mencakup resizing gambar, normalisasi, serta pembagian dataset menjadi training dan testing set dengan rasio 80:20.

Model CNN yang dikembangkan memiliki arsitektur yang terdiri dari empat lapisan konvolusi

dengan fungsi aktivasi ReLU, diikuti oleh lapisan pooling dan lapisan dense yang diakhiri dengan softmax untuk klasifikasi dua kelas [4]. Sistem ini diharapkan mampu memberikan deteksi dan klasifikasi kendaraan yang efektif dan efisien, yang pada akhirnya dapat meningkatkan kinerja dan keandalan sistem manajemen jalan tol. Pendekatan ini menawarkan potensi besar untuk meningkatkan sistem manajemen jalan tol dan memberikan kontribusi signifikan dalam bidang transportasi cerdas. Penelitian ini juga menegaskan bahwa penggunaan CNN dalam deteksi dan klasifikasi kendaraan dapat memberikan hasil yang sangat andal dan efisien, sehingga membuka jalan bagi aplikasi lebih lanjut dalam sistem transportasi dan pengelolaan lalu lintas.

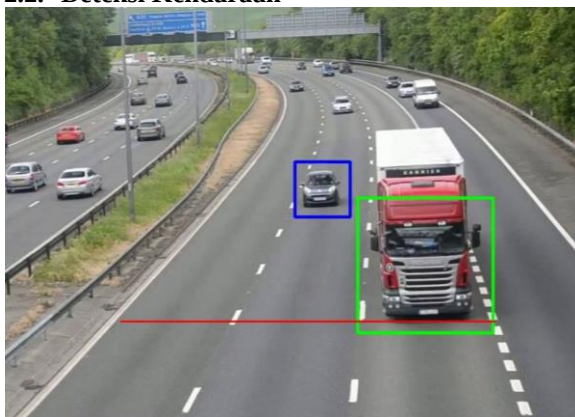
2. TINJAUAN PUSTAKA

2.1. Jalan Tol

Jalan tol adalah jalur transportasi umum yang memungut biaya tol dari penggunaannya dan berfungsi sebagai alternatif bagi jalan umum yang telah ada sebelumnya. Pengelolaan jalan tol dilakukan dengan maksud untuk mempercepat pengembangan jaringan jalan, di mana sebagian atau seluruh pendanaannya berasal dari pembayaran tol oleh pengguna, sehingga meringankan beban pemerintah. Pengelolaan jalan tol bertujuan untuk meningkatkan efisiensi dalam layanan distribusi dengan tujuan mendorong pertumbuhan ekonomi serta perencanaan pengembangan wilayah sesuai dengan rencana induk jaringan jalan [10].

Berdasarkan pada PP No. 15 Tahun 2005 tentang jalan, "jalan tol adalah jalan umum yang merupakan bagian dari sistem jaringan jalan dan sebagai jalan nasional yang penggunaannya diwajibkan membayar tol" [11].

2.2. Deteksi Kendaraan



Gambar 1. Deteksi Kendaraan

Deteksi kendaraan merupakan salah satu elemen krusial dalam sistem pemantauan lalu lintas otomatis. Teknologi ini dapat dimanfaatkan untuk berbagai keperluan, seperti pemantauan lalu lintas secara real-time, mengidentifikasi pelanggaran lalu lintas, serta mengoptimalkan aliran kendaraan [12]. Teknologi deteksi kendaraan menggunakan berbagai teknik pemrosesan citra dan algoritma pembelajaran mesin,

termasuk metode konvensional seperti latar belakang pengurangan, deteksi tepi, serta metode modern yang memanfaatkan jaringan saraf tiruan, terutama Convolutional Neural Network (CNN).

2.3. Klasifikasi Kendaraan

Klasifikasi kendaraan adalah proses pengkategorian jenis kendaraan berdasarkan karakteristik visualnya ke dalam beberapa kelas, seperti kendaraan ringan dan berat [13]. Tujuan utama dari klasifikasi ini adalah untuk mengenali jenis kendaraan yang melintas di jalan raya, yang dapat digunakan untuk analisis lalu lintas, penagihan tol elektronik, dan pengelolaan parkir. Teknik klasifikasi kendaraan sering memanfaatkan algoritma pembelajaran mesin yang memerlukan pelatihan pada dataset yang berisi gambar-gambar kendaraan dengan label yang sesuai.

2.4. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah jenis jaringan saraf tiruan yang dirancang untuk memproses data dalam bentuk grid, seperti gambar. CNN terdiri dari beberapa lapisan, termasuk lapisan konvolusi, pooling, dan fully connected, yang bekerja bersama untuk mengekstrak fitur dari gambar input dan melakukan klasifikasi. Lapisan konvolusi bertugas melakukan operasi konvolusi yang mengidentifikasi fitur-fitur penting seperti tepi, sudut, dan tekstur pada gambar. Lapisan pooling mengurangi dimensi fitur untuk mempercepat proses komputasi dan mengurangi kompleksitas model [14]. CNN telah terbukti sangat efektif dalam tugas-tugas pengenalan pola visual dan klasifikasi gambar.

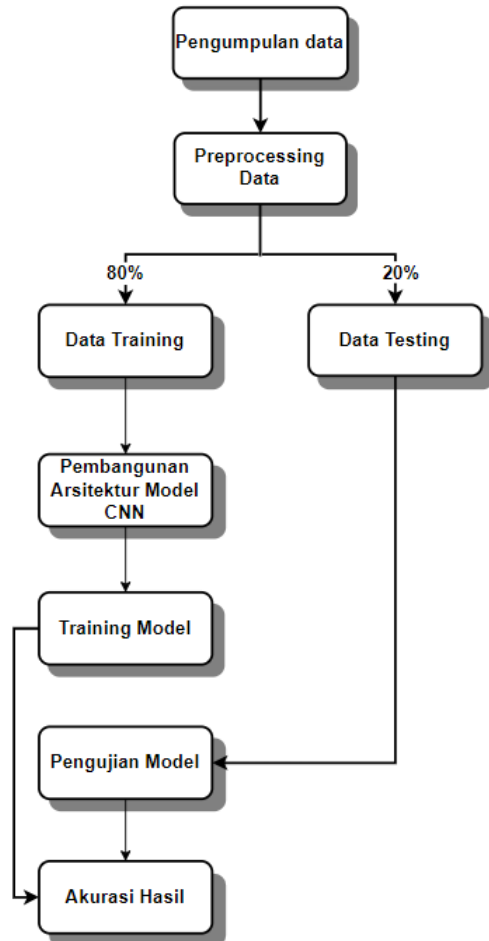
2.5. Computer Vision

Computer vision adalah teknologi yang menggabungkan kamera, komputasi edge, komputasi berbasis cloud, perangkat lunak, dan kecerdasan buatan (AI) untuk memungkinkan sistem "melihat" dan mengenali objek. Teknologi ini memungkinkan sistem untuk mengenali dan menganalisis objek dan orang dengan cepat, memeriksa hasil produksi, menganalisis demografi, dan melakukan berbagai tugas lain yang melibatkan pemrosesan visual [15]. Computer Vision digunakan dalam berbagai aplikasi, termasuk pengenalan pola, deteksi objek, segmentasi gambar, dan interpretasi konten visual. Dengan kemajuan dalam teknologi sensor dan kecerdasan buatan, Computer Vision telah menjadi bagian integral dari sistem otomasi industri, pengenalan wajah, kendaraan otonom, pengawasan keamanan, dan banyak lagi. Teknologi Computer Vision memainkan peran penting dalam pengembangan sistem deteksi dan klasifikasi kendaraan seperti yang dijelaskan dalam jurnal ini, dengan menyediakan alat dan metode untuk menganalisis data visual dengan akurat dan efisien.

3. METODE PENELITIAN

Penelitian ini memfokuskan pada pengembangan model arsitektur Convolutional Neural Network (CNN) yang dapat mengklasifikasikan jenis kendaraan

ringan dan kendaraan berat dengan tahapan penelitian dimulai dari pengumpulan data, pra-pemrosesan data, pembuatan model arsitektur, pelatihan model, hingga pengujian model. Secara keseluruhan, langkah-langkah metodologi penelitian yang dilakukan dapat dilihat pada Gambar 2. berikut:



Gambar 2. Flowchart Alur Metode Penelitian

3.1. Pengumpulan Data

Data yang dipakai pada penelitian ini berupa data gambar yang dibagi menjadi dua kelas, yaitu kelas kendaraan ringan dan kelas kendaraan berat. Data gambar terdiri dari 500 gambar kendaraan ringan dan 600 gambar kendaraan berat yang diperoleh melalui teknik web scraping dari berbagai sumber online.

3.2. Preprocessing Data

Proses pertama dalam preprocessing data adalah memuat gambar dari folder yang telah ditentukan. Gambar-gambar ini akan digunakan sebagai data latih dan data uji untuk model yang akan dibangun.

Setiap gambar diubah ukurannya menjadi 128x128 piksel untuk memastikan bahwa semua gambar memiliki dimensi yang sama. Konsistensi ini sangat penting untuk pemrosesan lebih lanjut. Mengurangi jumlah piksel juga dapat mengurangi kompleksitas komputasi dan memori. Setiap gambar diberi label sesuai dengan kategori foldernya

(misalnya, "kendaraan kecil" dengan label 0 dan "kendaraan berat" dengan label 1).

Selanjutnya, data gambar dan label digabungkan dan diacak untuk memastikan distribusi data yang merata dan mengurangi bias. Data kemudian dibagi menjadi training dan testing set dengan rasio 80:20 [16]. Normalisasi gambar dilakukan dengan mengatur nilai piksel gambar menjadi rentang 0-1 untuk meningkatkan performa model [17].

3.3. Pembangunan Arsitektur Model

Untuk keperluan klasifikasi gambar kendaraan ringan dan berat, model Convolutional Neural Network (CNN) dibangun dengan menggunakan Keras dari TensorFlow. Model terdiri dari beberapa lapisan konvolusi yang diikuti dengan lapisan pooling untuk ekstraksi fitur, serta lapisan dense untuk klasifikasi.

a. Conv2D

Lapisan konvolusi digunakan untuk mendeteksi berbagai fitur pada gambar. Setiap lapisan konvolusi menggunakan filter dengan ukuran 3x3 dan fungsi aktivasi ReLU.

b. MaxPooling2D

Lapisan pooling digunakan setelah setiap lapisan konvolusi untuk mengurangi dimensi data dan mengurangi jumlah parameter, yang membantu mencegah overfitting.

c. Flatten

Lapisan ini meratakan output dari lapisan sebelumnya menjadi satu dimensi, yang kemudian digunakan oleh lapisan dense.

d. Dense

Lapisan dense dengan 512 unit dan fungsi aktivasi ReLU digunakan untuk melakukan pemrosesan lebih lanjut setelah lapisan konvolusi.

e. Dropout

Lapisan dropout dengan dropout rate 0.5 digunakan untuk mencegah overfitting dengan mengabaikan setengah dari neuron selama proses pelatihan.

f. Output Layer

Lapisan dense terakhir dengan 2 unit (sesuai dengan jumlah kelas) dan fungsi aktivasi softmax digunakan untuk menghasilkan probabilitas kelas.

Model dikompilasi dengan optimizer Adam, dengan metrik akurasi. Optimizer Adam dipilih karena kemampuannya dalam menyesuaikan laju pembelajaran selama pelatihan, yang umumnya menghasilkan konvergensi yang lebih cepat dan stabil.

3.4. Pelatihan Model

Proses pelatihan ini terdiri dari beberapa iterasi, di mana model mempelajari pola dari data pelatihan dan dievaluasi menggunakan data validasi untuk mengukur performanya.

Parameter-parameter penting yang digunakan dalam pelatihan ini mencakup jumlah epochs, batch size, dan data validasi. Pada pelatihan ini, model dilatih selama 10 epochs, yang berarti seluruh dataset

digunakan untuk memperbarui bobot model sebanyak 10 kali. Selain itu, batch size yang digunakan adalah 32, menunjukkan jumlah sampel yang diproses dalam satu iterasi pembaruan bobot. Data validasi digunakan untuk mengevaluasi performa model pada setiap epoch, membantu memonitor potensi overfitting.

Selama evaluasi, metrik seperti akurasi dan loss dicatat, yang memberikan gambaran tentang kinerja model pada data yang belum pernah dilihat sebelumnya

3.5. Pengujian Model

Langkah berikutnya adalah menguji dan mengevaluasi model. Evaluasi model dilakukan untuk menilai performanya pada data yang belum pernah dilihat sebelumnya, dengan menggunakan dataset pengujian yang telah dipisahkan dari dataset awal. Evaluasi ini penting untuk memahami sejauh mana model dapat menggeneralisasi dan memberikan prediksi yang akurat pada data baru.

Proses evaluasi dilakukan dengan menggunakan metode evaluate dari Keras, yang menghitung nilai loss dan akurasi model pada dataset pengujian (X_{test} dan y_{test}).

4. HASIL DAN PEMBAHASAN

Pada bagian ini, kami akan menguraikan hasil dari pelatihan dan pengujian model Convolutional Neural Network (CNN) yang dikembangkan untuk klasifikasi kendaraan ringan dan kendaraan berat. Proses ini dilakukan menggunakan Google Colaboratory dengan bahasa pemrograman Python. Hasil akan dievaluasi berdasarkan akurasi model, fungsi loss, serta metrik evaluasi lainnya seperti precision, recall, f1-score, dan confusion matrix.

4.1. Dataset

Penelitian ini memanfaatkan dataset gambar kendaraan yang diperoleh melalui teknik web scraping dari berbagai sumber online. Dataset ini terdiri dari dua kategori utama: kendaraan ringan dan kendaraan berat. Total dataset yang didapat mencakup 1,100 gambar, yang terdiri dari 500 gambar kendaraan ringan dan 600 gambar kendaraan berat.

Tabel 1. Keseluruhan Dataset

Dataset	Jumlah Data
Data Kendaraan Ringan	500
Data Kendaraan Berat	600

Dataset dibagi menjadi data latih dan data uji dengan rasio 80:20. Pembagian ini dilakukan untuk memastikan model dapat diuji pada data yang belum pernah dilihat sebelumnya, sehingga memberikan evaluasi kinerja yang lebih akurat. Sebanyak 836 gambar digunakan untuk melatih model, dan 209 gambar digunakan untuk menguji model. Pembagian ini memastikan bahwa model memiliki cukup data untuk belajar serta memberikan evaluasi yang baik pada data yang tidak terlibat dalam proses pelatihan.

Tabel 2. Pembagian Dataset

Jenis Kelas	Data Latih	Data Uji
Kendaraan ringan & berat	836	200

4.2. Pengembangan Arsitektur Model

Model CNN yang dikembangkan untuk klasifikasi kendaraan ringan dan berat memiliki arsitektur sebagai berikut:

- Lapisan Konvolusi (Conv2D)**
Empat lapisan konvolusi digunakan untuk mengekstrak fitur dari gambar. Setiap lapisan konvolusi menggunakan filter ukuran 3x3 dengan fungsi aktivasi ReLU (Rectified Linear Unit). Fungsi ReLU digunakan untuk memperkenalkan non-linearitas ke dalam model, sehingga mampu menangkap fitur yang lebih kompleks.
- Lapisan Pooling (MaxPooling2D)**
Setelah setiap lapisan konvolusi, lapisan pooling digunakan untuk mengurangi dimensi fitur dan mencegah overfitting. Lapisan pooling yang digunakan adalah MaxPooling2D dengan ukuran 2x2.
- Lapisan Flatten**
Lapisan ini meratakan output dari lapisan konvolusi dan pooling menjadi satu dimensi, yang kemudian digunakan oleh lapisan dense.
- Lapisan Dense**
Lapisan dense dengan 512 unit dan fungsi aktivasi ReLU digunakan untuk melakukan pemrosesan lebih lanjut setelah lapisan konvolusi. Lapisan dense ini bertujuan untuk melakukan klasifikasi berdasarkan fitur-fitur yang telah diekstraksi.
- Lapisan Dropout**
Lapisan dropout dengan dropout rate 0.5 digunakan untuk mencegah overfitting dengan mengabaikan setengah dari neuron selama proses pelatihan.
- Lapisan Output**
Lapisan dense terakhir dengan 2 unit (sesuai dengan jumlah kelas) dan fungsi aktivasi softmax digunakan untuk menghasilkan probabilitas kelas

Dibawah ini adalah gambar hasil ringkasan model yang telah dibuat.

Model: "sequential_15"

Layer (type)	Output Shape	Param #
conv2d_44 (Conv2D)	(None, 126, 126, 32)	896
max_pooling2d_44 (MaxPooling2D)	(None, 63, 63, 32)	0
conv2d_45 (Conv2D)	(None, 61, 61, 64)	18,496
max_pooling2d_45 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_46 (Conv2D)	(None, 28, 28, 128)	73,856
max_pooling2d_46 (MaxPooling2D)	(None, 14, 14, 128)	0
conv2d_47 (Conv2D)	(None, 12, 12, 128)	147,584
max_pooling2d_47 (MaxPooling2D)	(None, 6, 6, 128)	0
flatten_12 (Flatten)	(None, 4608)	0
dense_30 (Dense)	(None, 512)	2,359,808
dropout_14 (Dropout)	(None, 512)	0
dense_31 (Dense)	(None, 2)	1,026

Total params: 2,601,666 (9.92 MB)
Trainable params: 2,601,666 (9.92 MB)
Non-trainable params: 0 (0.00 B)

Gambar 3. Hasil Ringkasan Model

4.3. Pelatihan Model

Proses pelatihan model dilakukan dengan langkah-langkah sebagai berikut:

a. Inisialisasi Model

Model CNN diinisialisasi dengan arsitektur yang telah dijelaskan sebelumnya.

b. Kompilasi Model

Model dikompilasi menggunakan optimizer Adam, dengan loss function categorical cross-entropy dan metrik akurasi. Optimizer Adam dipilih karena kemampuannya dalam menyesuaikan laju pembelajaran selama pelatihan, yang umumnya menghasilkan konvergensi yang lebih cepat dan stabil.

c. Pelatihan Model

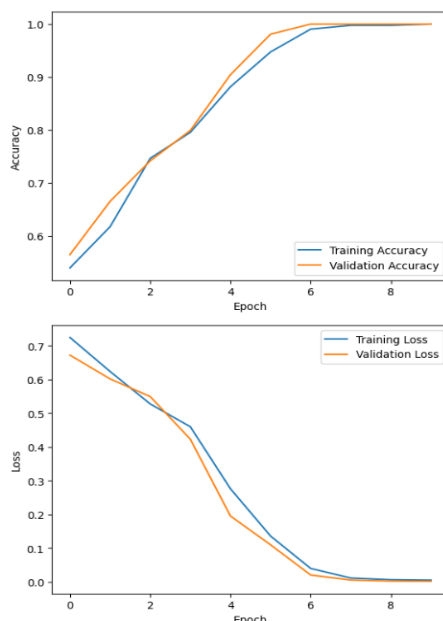
Model dilatih menggunakan data latih (836 gambar) selama 10 epoch dengan batch size 32. Selama pelatihan, model dievaluasi menggunakan data validasi untuk mengukur performa dan mendeteksi overfitting.

4.4. Hasil Pelatihan Model dan Validasi Model

Model CNN yang dikembangkan dilatih selama 10 epoch dengan batch size 32. Selama proses pelatihan, model dievaluasi menggunakan data validasi untuk mengukur performa dan mendeteksi overfitting. Grafik di bawah ini menunjukkan perubahan akurasi dan loss pada data pelatihan dan validasi selama epoch.

Epoch 1/10	7s 218ms/step - accuracy: 0.5186 - loss: 0.7909 - val_accuracy: 0.5646 - val_loss: 0.6724
Epoch 2/10	7s 242ms/step - accuracy: 0.5798 - loss: 0.6483 - val_accuracy: 0.6651 - val_loss: 0.6819
Epoch 3/10	7s 247ms/step - accuracy: 0.7396 - loss: 0.5651 - val_accuracy: 0.7416 - val_loss: 0.5495
Epoch 4/10	7s 249ms/step - accuracy: 0.7665 - loss: 0.4890 - val_accuracy: 0.7990 - val_loss: 0.4230
Epoch 5/10	7s 256ms/step - accuracy: 0.8646 - loss: 0.3293 - val_accuracy: 0.9043 - val_loss: 0.3951
Epoch 6/10	7s 253ms/step - accuracy: 0.9466 - loss: 0.1492 - val_accuracy: 0.9809 - val_loss: 0.3897
Epoch 7/10	7s 244ms/step - accuracy: 0.9866 - loss: 0.0658 - val_accuracy: 1.0000 - val_loss: 0.0201
Epoch 8/10	7s 265ms/step - accuracy: 0.9958 - loss: 0.0166 - val_accuracy: 1.0000 - val_loss: 0.0052
Epoch 9/10	7s 245ms/step - accuracy: 0.9990 - loss: 0.0046 - val_accuracy: 1.0000 - val_loss: 0.0020
Epoch 10/10	7s 248ms/step - accuracy: 1.0000 - loss: 0.0040 - val_accuracy: 1.0000 - val_loss: 0.0015

Gambar 4. Hasil Training Model



Gambar 5. Grafik akurasi dan loss selama proses pelatihan dan validasi.

Dari Gambar 5, terlihat bahwa akurasi model meningkat dan loss menurun seiring bertambahnya epoch. Akurasi pelatihan meningkat dari 51.86% pada epoch pertama menjadi 100% pada epoch kesepuluh, sementara akurasi validasi meningkat dari 56.46% menjadi 100% pada epoch kesepuluh. Penurunan loss juga terlihat jelas, dari 0.7909 pada epoch pertama menjadi 0.0015 pada epoch kesepuluh. Hal ini menunjukkan bahwa model belajar dengan baik dari data pelatihan.

Namun, perbedaan yang signifikan antara akurasi pelatihan dan validasi pada beberapa epoch menunjukkan bahwa model mencapai akurasi 100% pada data validasi lebih cepat dibandingkan dengan data pelatihan, yang dapat mengindikasikan potensi overfitting. Model tampaknya sangat baik dalam mengklasifikasikan data pelatihan, tetapi perlu diperhatikan untuk memastikan generalisasi yang baik pada data yang tidak terlihat sebelumnya.

4.5. Pengujian Model

Setelah proses pelatihan selesai, model diuji menggunakan data uji (209 gambar) yang telah dipisahkan dari dataset awal. Evaluasi model dilakukan untuk menilai performanya pada data yang belum pernah dilihat sebelumnya, dengan menggunakan berbagai metrik evaluasi seperti akurasi, precision, recall, f1-score, dan confusion matrix.

4.5.1. Akurasi

Akurasi mengukur proporsi prediksi yang benar dari keseluruhan data uji [18]. Rumus untuk menghitung akurasi adalah:

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN}$$

Gambar 6. Rumus Akurasi

Di mana:

- TP (True Positive):** Jumlah data positif yang diklasifikasikan dengan benar.
- TN (True Negative):** Jumlah data negatif yang diklasifikasikan dengan benar.
- FP (False Positive):** Jumlah data negatif yang diklasifikasikan salah sebagai positif.
- FN (False Negative):** Jumlah data positif yang diklasifikasikan salah sebagai negatif.

4.5.2. Precision

Precision mengukur proporsi data yang diklasifikasikan sebagai positif yang benar-benar positif [18]. Precision dihitung dengan rumus:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Gambar 7. Rumus Precision

Precision yang tinggi menunjukkan bahwa model memiliki tingkat false positive yang rendah.

4.5.3. Recall

Recall, juga dikenal sebagai sensitivitas atau true positive rate, mengukur proporsi data positif yang berhasil diklasifikasikan dengan benar [18]. Recall dihitung dengan rumus:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Gambar 8. Rumus Recall

Recall yang tinggi berarti model berhasil mengidentifikasi sebagian besar data positif.

4.5.4. F1-Score

F1-score adalah rata-rata harmonik dari precision dan recall, yang memberikan keseimbangan antara kedua metrik ini [19]. Rumus untuk menghitung F1-score adalah:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Gambar 9. Rumus Akurasi

F1-score sangat berguna ketika terdapat ketidakseimbangan kelas dalam dataset.

4.5.5. Confusion Matrix

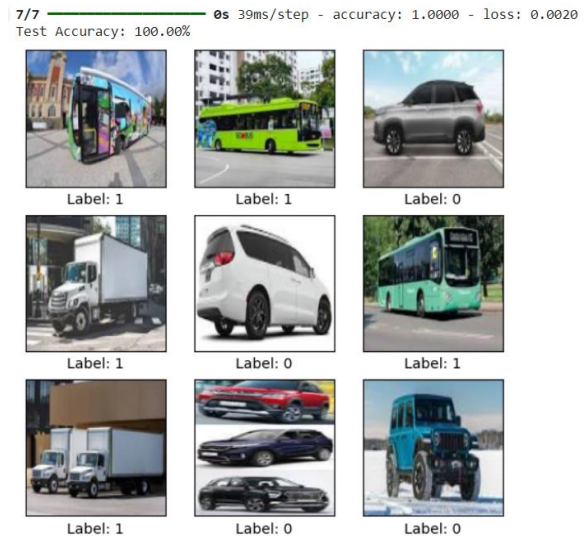
Confusion Matrix adalah alat evaluasi yang digunakan untuk menilai kinerja model klasifikasi dengan memvisualisasikan performa model dalam bentuk matriks [20]. Matriks ini memberikan informasi tentang hasil klasifikasi yang benar dan salah, serta mendetailkan bagaimana setiap kelas diprediksi oleh model. Dalam konteks penelitian ini, Confusion Matrix digunakan untuk menganalisis hasil deteksi dan klasifikasi kendaraan ringan dan berat oleh model Convolutional Neural Network (CNN). Matriks ini membantu untuk memahami secara mendetail bagaimana model mengklasifikasikan kendaraan dalam dua kategori: kendaraan ringan dan kendaraan berat.

	Kendaraan Ringan	Kendaraan Berat
Kendaraan Ringan	TP	FN
Kendaraan Berat	FP	TN

Gambar 10. Contoh Confusion Matrix

- True Positives (TP):** Jumlah kendaraan ringan yang benar-benar diklasifikasikan sebagai kendaraan ringan.
- False Negatives (FN):** Jumlah kendaraan ringan yang salah diklasifikasikan sebagai kendaraan berat.
- False Positives (FP):** Jumlah kendaraan berat yang salah diklasifikasikan sebagai kendaraan ringan.

- True Negatives (TN):** Jumlah kendaraan berat yang benar-benar diklasifikasikan sebagai kendaraan berat.



Gambar 11. Hasil Klasifikasi Gambar yang Diuji dan Diberi Label

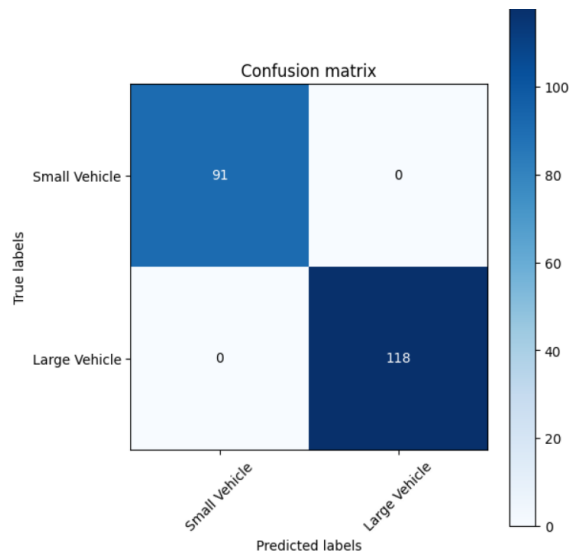
Setelah pengujian, model berhasil mencapai akurasi 100% pada data uji, seperti yang ditunjukkan dalam Gambar 11. Hal ini menunjukkan bahwa model tidak hanya efektif dalam mengklasifikasikan data latih dan data validasi, tetapi juga sangat akurat dalam mengklasifikasikan data uji yang belum pernah dilihat sebelumnya.

Untuk mengevaluasi kinerja model pada data uji, kami menggunakan berbagai metrik evaluasi. Hasil evaluasi disajikan dalam Gambar 12. di bawah ini.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	91
1	1.00	1.00	1.00	118
accuracy			1.00	209
macro avg	1.00	1.00	1.00	209
weighted avg	1.00	1.00	1.00	209

Gambar 12. Hasil evaluasi model pada data uji

Gambar 12. menunjukkan bahwa model CNN yang dikembangkan memiliki akurasi sangat tinggi, dengan precision, recall, dan f1-score mencapai nilai maksimum. Hal ini menunjukkan bahwa model memiliki kinerja yang sangat baik secara keseluruhan.



Gambar 13. Confusion Matrix model CNN pada data uji

Confusion matrix memberikan gambaran yang lebih detail tentang performa model dengan menunjukkan jumlah prediksi benar dan salah untuk masing-masing kelas. Gambar 13. menunjukkan bahwa model mampu mengklasifikasikan sebagian besar kendaraan ringan dan berat dengan benar. Dari 209 kendaraan, 91 kendaraan ringan diklasifikasikan dengan benar, sementara 0 kendaraan ringan diklasifikasikan salah sebagai kendaraan berat. Sebaliknya, 118 kendaraan berat diklasifikasikan dengan benar, dan 0 kendaraan berat diklasifikasikan salah sebagai kendaraan ringan. Dengan demikian, model ini menunjukkan kinerja yang sempurna pada dataset ini, dengan akurasi 100% untuk kedua kelas kendaraan. Model berhasil membedakan antara Kendaraan Kecil dan Kendaraan Besar tanpa membuat kesalahan dalam skenario uji ini.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengembangkan model Convolutional Neural Network (CNN) untuk deteksi dan klasifikasi kendaraan ringan dan berat dengan hasil yang sangat memuaskan, di mana model mencapai akurasi training 100%, akurasi validasi 100%, dan akurasi testing 100%, serta precision, recall, dan f1-score yang sempurna. Meski demikian, terdapat potensi overfitting dan keterbatasan dalam variasi dataset yang digunakan, sehingga disarankan untuk pengembangan selanjutnya melibatkan teknik augmentasi data, pengujian pada dataset yang lebih besar dan bervariasi, serta eksplorasi arsitektur CNN yang lebih kompleks atau metode transfer learning untuk meningkatkan generalisasi dan keandalan model dalam aplikasi sistem manajemen lalu lintas.

DAFTAR PUSTAKA

[1] P.K. Sukanpio, "Strategi Manajemen Dan Rekayasa Lalu Lintas Di Ruas Jalan Jenderal Sudirman Kota Kupang Dengan Menggunakan Analisis SWOT," Universitas Katolik Widya

Mandira, 06 Mei 2022. [Online]. Tersedia: <http://repository.unwira.ac.id/5188/2/BAB%20I.pdf> [Diakses: 28 Mei 2024].

- [2] W. Andriyani, R. Purnomo, S.A. Hendrawan, A.I. Irvani, A. Sujarwo, Y.N. Asri, A.H.S. Jones, N.R. Feta, I.Y. Wulandari, S. Hatma, H.Y. Suhendi, A.Y.P. Yusuf, D. Haryadi, H. Lubis, "Data Sebagai Fondasi Kecerdasan Buatan," [Online]. Tersedia: [https://books.google.com/books?hl=id&lr=&id=GLgFEQAAQBAJ&oi=fnd&pg=PP1&dq=Kemajuan+dalam+bidang+kecerdasan+buatan+\(AI\)+dan+pembelajaran+mesin+\(ML\)+telah+membuka+peluang+baru+untuk+pengembangan+sistem+deteksi+dan+klasifikasi+kendaraan+yang+lebih+canggih.&ots=U0fDqKSfKD&sig=sn_ahIzRMAtofw-OKMWTtLfQIjE](https://books.google.com/books?hl=id&lr=&id=GLgFEQAAQBAJ&oi=fnd&pg=PP1&dq=Kemajuan+dalam+bidang+kecerdasan+buatan+(AI)+dan+pembelajaran+mesin+(ML)+telah+membuka+peluang+baru+untuk+pengembangan+sistem+deteksi+dan+klasifikasi+kendaraan+yang+lebih+canggih.&ots=U0fDqKSfKD&sig=sn_ahIzRMAtofw-OKMWTtLfQIjE) [Diakses: 29 Mei 2024].
- [3] "Convolutional Neural Networks (CNN) Overview," Encord Blog, 24 Juli 2023. [Online]. Tersedia: <https://encord.com/blog/convolutional-neural-networks-explained/> [Diakses: 29 Mei 2024].
- [4] L.N. Lukman, "Penerapan Convolutional Neural Network (CNN) dalam Klasifikasi Alat Kesehatan," Medium, 16 Mei 2024. [Online]. Tersedia: <https://medium.com/@lianurlianalukman/penerapan-convolutional-neural-network-cnn-dalam-klasifikasi-alat-kesehatan-3e3e4a9f8e58> [Diakses: 29 Mei 2024].
- [5] F. Hafifah, S. Rahman, dan M. S. Asih, "Klasifikasi Jenis Kendaraan Pada Jalan Raya Menggunakan Metode Convolutional Neural Networks (CNN)," TIN: Terapan Informatika Nusantara, vol. 2, no. 5, pp. 292-301, 2021.
- [6] D. Iswanto dan D. Handayani UN, "Klasifikasi Penyakit Tanaman Jagung Menggunakan Metode Convolutional Neural Network (CNN)," Jurnal Ilmiah Universitas Batanghari Jambi (JIUBJ), vol. 22, no. 2, pp. 900-905, 2022.
- [7] F. N. Cahya, N. Hardi, D. Riana, dan S. Hadiyanti, "Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN)," SISTEMASI: Jurnal Sistem Informasi, vol. 10, no. 3, pp. 618-626, 2021.
- [8] R. R. Karim and A. Herlangga, "Implementasi Klasifikasi Senjata Tradisional Jawa Barat Menggunakan Convolutional Neural Network (CNN) dengan Metode Transfer Learning," JITET (Jurnal Informatika dan Teknik Elektro Terapan), vol. 12, no. 2, pp. 1210-1216, 2023, doi: 10.23960/jitet.v12i2.4166.
- [9] M. E. Prasetyo, M. R. Faza, R. Pratama, S. N. H. Alhabsy, H. Purwanti, dan A. P. A. Masa, "Klasifikasi Ragam Kendaraan Menggunakan Metode Convolutional Neural Network (CNN),"

- Adopsi Teknologi dan Sistem Informasi (ATASI), vol. 2, no. 2, 2023.
- [10] M. M. Sembiring, "Analisis Dampak Pembangunan Infrastruktur Terhadap Pendapatan Masyarakat: Studi Kasus Pembangunan Jalan Tol Pekanbaru - Minas," Repository Universitas Islam Riau, 15 Sep 2022. [Online]. Tersedia: <https://repository.uir.ac.id/15148/> [Diakses 31 Mei 2024].
- [11] "PERATURAN PEMERINTAH REPUBLIK INDONESIA," JDIH Kemenkeu. [Online]. Tersedia: <https://jdih.kemenkeu.go.id/FullText/2005/15TAHUN2005PP.HTM>. [Diakses: 02 September 2024].
- [12] A. Pamungkas, "Background Subtraction Dalam Ruang Warna HSV," Pemrograman Matlab, 7 September 2023. [Online]. Tersedia: <https://pemrogramanmatlab.com/2023/09/07/background-subtraction-dalam-ruang-warna-hsv-2/#more-10286>. [Diakses: 02 September 2024].
- [13] "What is Vehicle Classification?," isarsoft, 1 Juni 2024. [Online]. Tersedia: <https://www.isarsoft.com/knowledge-hub/vehicle-classification>. [Diakses: 02 September 2024].
- [14] R. Z. Lini, "Klasifikasi Produk menggunakan Convolutional Neural Network (CNN)," Medium, 16 Mei 2024. [Online]. Tersedia: <https://medium.com/@ranizatalini/klasifikasi-produk-menggunakan-convolutional-neural-network-cnn-163bd2439fb8> [Diakses 31 Mei 2024].
- [15] "Apa itu Computer Vision?," intel. [Online]. Tersedia: <https://www.intel.co.id/content/www/id/id/internet-of-things/computer-vision/overview.html>. [Diakses: 02 September 2024].
- [16] R.A. Saputra, D.R.R. Putra, and M.A. Asyrofi, "Implementasi Convolutional Neural Network (Cnn) Untuk Mendeteksi Penggunaan Masker Pada Gambar," JITET (Jurnal Informatika dan Teknik Elektro Terapan), vol. 11, no. 3, hal. 710-714, 2023, doi: 10.23960/jitet.v11i3.3286
- [17] M. Rahman, Asriyanik, and A. Pambudi, "Implementasi Algoritma Convolutional Neural Network (CNN) Untuk Identifikasi Kualitas Tanaman Selada Berdasarkan Citra Daun," JITET (Jurnal Informatika dan Teknik Elektro Terapan), vol. 11, no. 3, hal. 851-858, 2023, doi: 10.23960/jitet.v11i3s1.3368.
- [18] "Accuracy, precision, and recall in multi-class classification," Evidently AI, [Online]. Tersedia: <https://www.evidentlyai.com/classification-metrics/multi-class-metrics>. [Diakses: 02 September 2024].
- [19] S. Setiawan, "Membicarakan Precision, Recall, dan F1-Score," medium, 12 Juli 2020. [Online]. Tersedia: <https://stevkarta.medium.com/membicarakan-precision-recall-dan-f1-score-e96d81910354>. [Diakses: 02 September 2024].
- [20] N. A. Ahmed, "What is A Confusion Matrix in Machine Learning? The Model Evaluation Tool Explained," DataCamp, Nov 2023. [Online]. Tersedia: <https://www.datacamp.com/tutorial/what-is-a-confusion-matrix-in-machine-learning>. [Diakses: 02 September 2024].