

Analisis Perbandingan Evaluasi Metode *Deep Learning* pada Klasifikasi Jenis Kendaraan

Anggay Luri Pramana

*Program Studi Teknik Informatika, Fakultas Ilmu Komputer,
Universitas Nahdlatul Ulama Sidoarjo
Jln. Mongonsidi Kav. DPR Sidoklumpuk Sidoarjo
luri409.tif@unusida.ac.id*

Abstrak

Pengenalan kendaraan memiliki kompleksitas yang tinggi, permasalahan yang dihadapi saat melakukan penelitian kendaraan seperti variasi jenis kendaraan, kondisi pencahayaan, perspektif, resolusi, kualitas gambar serta warna dan tekstur menjadi faktor utama. Permasalahan ini memerlukan pendekatan multidisiplin ilmu dengan kombinasi teknologi pengolahan citra, machine learning, dan pengenalan pola. Pendekatan inovatif dan penelitian secara konsisten penting untuk meningkatkan kinerja sistem serta mencoba segala model arsitektur deep learning yang telah dikembangkan.

Penelitian ini bertujuan untuk membandingkan Model Neural Network untuk klasifikasi kendaraan kelas 1 hingga kelas 5 berdasarkan jenis klasifikasi pada jalan tol. Model yang dibandingkan dalam penelitian ini adalah Convolutional Neural Network, ResNet50, dan VGG16. Model akan diuji dengan gambar masukan seluruh bagian kendaraan setelah itu gambar masukan diubah ukurannya menjadi 224x224 untuk setiap gambar masukan. Skenario dilakukan dengan menggunakan 75 epoch pada setiap model dengan total 500 data untuk setiap kelompok dan setiap kelompok. Persentase data latih dengan data uji adalah 80% data latih dan 20% data uji. Jumlah kelompok kelas ada 5 yaitu Kelompok 1, Kelompok 2, Kelompok 3, Kelompok 4, dan Kelompok 5.

Model VGG16 mendapatkan akurasi tertinggi dengan akurasi 91%, Convolutional Neural Network 84% dan ResNet50 mendapatkan akurasi 74%. Hasil yang diperoleh menunjukkan efektivitas model VGG16 lebih tinggi terhadap CNN dan ResNet50. Dengan demikian penelitian ini dapat memberikan persepsi yang berguna untuk penelitian selanjutnya guna meningkatkan pengambilan gambar dengan kamera yang lebih berkualitas guna meningkatkan sistem transportasi cerdas.

Kata kunci— *Convolutional Neural Network, Deep learning, ResNet50, VGG16, Smart Transportation System*

Abstract

Vehicle recognition has high complexity, and problems that arise when conducting vehicle research such as variations in vehicle type, lighting conditions, perspective, resolution, image quality, and color and texture are the main factors. This problem requires a multidisciplinary approach with a combination of image processing technology, machine learning, and pattern recognition. Innovative approaches and research are consistently important to improve system performance as well as trying out all the deep learning architecture models that have been developed.

This research aims to compare Neural Network Models for class 1 to class 5 vehicle classification based on the type of classification on toll roads. The models compared in this research are Convolutional Neural Network, ResNet50, and VGG16. The model will be tested with input images of all parts of the vehicle after which the input images are resized to 224x224 for each input image. The scenario was carried out using 75 epochs in each model with a total of

500 data for each group and each group. The percentage of training data and test data is 80% training data and 20% test data. There are 5 class groups, namely Group 1, Group 2, Group 3, Group 4, Group 5.

The VGG16 model got the highest accuracy with 91% accuracy, Convolutional Neural Network 84% and ResNet50 got 74% accuracy. The results obtained show that the effectiveness of the VGG16 model is higher against CNN and ResNet50. Thus, this research can provide useful perceptions for further research to improve image capture with higher-quality cameras to improve intelligent transportation systems.

Keywords— Convolutional Neural Network, Deep learning, ResNet50, VGG16, Smart Transportation System

1. PENDAHULUAN

Transportasi adalah perpindahan orang atau barang dari suatu tempat ke tempat lain dengan menggunakan kendaraan yang digerakkan oleh manusia atau mesin. Transportasi digunakan untuk memudahkan masyarakat dalam melakukan aktivitas sehari-hari. Transportasi sendiri terbagi menjadi 3 yaitu, darat, udara dan air. Transportasi di Indonesia pada sektor darat mempunyai jenis kelas kendaraan yang beragam khususnya pada jalan tol (Pramana et al., 2020). berdasarkan informasi yang dikutip dari situs Kementerian Pekerjaan Umum dan Perumahan Rakyat. Jenis kendaraan dibagi menjadi 5 kategori yaitu golongan 1 hingga golongan 5.

Di era teknologi saat ini, banyak inovasi di bidang teknologi yang dikembangkan dalam bidang sistem transportasi. Klasifikasi kendaraan menggunakan pemrosesan gambar dan visi komputer memberikan kemudahan transportasi. Perasaan nyaman di jalan padat sehingga dapat menghitung kendaraan masuk dan keluar (Aug et al., 2015). *Intelligent Vehicles Symposium IEEE* adalah forum teknis tahunan besar yang disponsori oleh IEEE *Intelligent Transportation Systems Society* (ITSS), yang mempertemukan para peneliti dan praktisi di seluruh dunia untuk berbagi kemajuan terkini dalam teori dan teknologi terkait *smart transportation*. Pada penelitian tahun 2017 dilakukan dengan menggunakan metode CNN untuk klasifikasi kendaraan (Chen et al., 2017) dan juga dilakukan klasifikasi kendaraan dengan menggunakan metode CNN (Huang et al., 2017).

Penelitian tentang pengenalan kendaraan mendorong inovasi dalam bidang teknologi pengolahan citra, *artificial intelligence*, dan *machine learning*. Data yang telah di kumpulkan dari pengenalan kendaraan dapat digunakan untuk berbagai analisa data dan identifikasi. Identifikasi kendaraan yang dilakukan manual atau semi-otomatis menggunakan cara konvensional dengan teknik pengolahan citra, ekstraksi fitur, deteksi atau segmentasi citra. Pada penelitian ini mekanisme identifikasi kendaraan di lakukan dengan dengan cara otomatis dengan memanfaatkan metode *deep learning*. Dalam metode *deep learning* terdapat *feature learning* dan *fine tuning* yang dapat mempelajari pola dari kendaraan secara bertahap dari *low-level feature* hingga *high-level feature*. Dengan munculnya *deep learning* representasi fitur yang kompleks dari data mentah dapat di pelajari tanpa perlu ekstraksi fitur manual.

Convolutional Neural Network (CNN) merupakan metode yang digunakan untuk data berupa gambar. CNN dapat digunakan untuk mendeteksi dan mengenali objek gambar. Arsitektur *Neural Network* memiliki kemampuan jaringan *multilayer* dengan penurunan gradien untuk mempelajari pemetaan nonlinier berdimensi tinggi yang kompleks (Le Callet et al., 2006). CNN memiliki *neuron* yang memiliki bobot, bias, dan fungsi aktivasi. Secara garis besar CNN memiliki dua tahapan yaitu: *Feature Learning*

dan *Classification*. *Convolutional Neural Network* (CNN) merupakan pengembangan dari *multilayer perceptron* (MLP) dirancang untuk mengolah data dua dimensi dalam bentuk gambar (Le Callet et al., 2006). CNN termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringannya tinggi dan banyak diterapkan pada data gambar. Klasifikasi gambar dasar dapat digunakan dengan MLP, namun metode MLP kurang cocok karena tidak cocok untuk menyimpan informasi spasial dari data gambar, dan menganggap setiap piksel merupakan fitur independen memberikan hasil yang buruk.

Klasifikasi jenis kendaraan adalah tugas visi komputer umum yang melibatkan penentuan kategori atau jenis kendaraan berdasarkan gambar masukan. Tugas ini memiliki berbagai penerapan praktis, termasuk pemantauan lalu lintas, pengawasan, dan mengemudi otonom. *Convolutional Neural Network* (CNN) telah terbukti sangat efektif untuk tugas klasifikasi gambar, dan arsitektur seperti ResNet50 dan VGG16 adalah pilihan populer karena kemampuannya mengekstrak fitur kompleks dari gambar. Dalam pendahuluan ini, peneliti akan mendalami konsep klasifikasi jenis kendaraan menggunakan CNN, ResNet50, dan VGG16.

CNN adalah kelas *deep neural networks* yang dirancang khusus untuk memproses data grid terstruktur, seperti gambar. Mereka terdiri dari beberapa lapisan, termasuk *convolutional layers*, *pooling layers*, dan *fully connected layers*. CNN sangat cocok untuk tugas-tugas seperti klasifikasi gambar karena mereka dapat secara otomatis mempelajari dan mengekstrak fitur yang relevan dari gambar, sehingga mengurangi kebutuhan rekayasa fitur manual. ResNet (*Residual Network*) adalah arsitektur jaringan saraf dalam yang dikenal karena kemampuannya untuk melatih *deep neural network* secara efektif. ResNet50 adalah varian spesifik yang mencakup 50 lapisan, menggunakan *residual connections* (*skip connections*) untuk mengatasi masalah gradien yang hilang. Arsitektur ini telah menunjukkan performa luar biasa dalam berbagai tugas klasifikasi gambar, termasuk klasifikasi jenis kendaraan. VGG (*Visual Geometry Group*) adalah arsitektur jaringan saraf konvolusional dalam lainnya yang menjadi runner-up dalam *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) pada tahun 2014. VGG16 adalah varian dari arsitektur VGG yang terdiri dari 16 lapisan bobot, termasuk 13 *convolutional layers* dan 3 *fully connected layers*. VGG16 dikenal karena kesederhanaan dan kemudahannya (Naveen & Diwan, 2021).

2. METODE PENELITIAN

2.1 Pengambilan Data

Pengambilan data dilakukan dengan membagi menjadi kategori data yang berisikan pengelompokan setiap kelasnya. Teknik pengambilan sampel dijelaskan mengenai pemisahan presentase data *training*, data validasi, dan data *testing*.



Gambar 1. Daftar kelas setiap golongan

2.1.1 Kategori Data

Jumlah *dataset* sebanyak 500 data dengan masing-masing kelas 100 data. Kelas atau labelnya terbagi menjadi 5 yaitu golongan 1, golongan 2, golongan 3, golongan 4 dan golongan 5. Pembagian setiap kelas dapat dilihat pada gambar 1.

2.1.2 Teknik Pengambilan Sampel

Pemisahan kumpulan data yang terdiri dari 500 gambar menjadi kumpulan pelatihan 80% dan kumpulan pengujian 20% untuk pra pemrosesan gambar melibatkan pembagian data menjadi dua sub kumpulan. Proses dimulai dengan mengacak urutan 500 gambar secara acak. Langkah ini memastikan bahwa gambar berada dalam urutan acak. Setelah itu besar kecilnya set pelatihan dan pengujian ditentukan berdasarkan rasio 80 – 20. Pada 80% set pelatihan dan 20% :

- a. *Training set size*: $0,80 \times 500 = 400$ gambar
- b. *Testing set size*: $0,20 \times 500 = 100$ gambar

Setelah itu ambil 400 gambar pertama dari data set secara acak sebagai *training set*. Ini akan digunakan untuk *preprocessing* dan *training* model. Kemudian mengambil 100 gambar sisanya sebagai *testing set*, yang akan digunakan untuk mengetahui performa model setelah pra pemrosesan dan pelatihan.

2.2 Convolutional Neural Network

Pertama, gambar masukan direpresentasikan sebagai nilai piksel. Pada gambar *grayscale*, setiap piksel memiliki satu nilai intensitas. Sedangkan pada gambar berwarna (RGB), setiap piksel biasanya memiliki tiga nilai (Merah, Hijau, dan Biru). Kemudian *filter* atau *kernel* berupa kotak kecil berisi bobot-bobot yang bisa dipelajari serta direpresentasikan sebagai *grid*. Selama operasi konvolusi, *filter* ini melintasi gambar masukan (Khagi & Kwon, 2021). Selanjutnya dilakukan langkah konvolusi pada setiap posisi penempatan *filter* pada gambar masukan, dilakukan operasi konvolusi. Operasi ini melibatkan mengalikan nilai *filter* berdasarkan elemen dengan nilai yang sesuai di *patch* gambar masukan (Pang et al., 2018).

Kemudian dilakukan perkalian setiap elemen piksel, hasilnya dijumlahkan hingga menghasilkan nilai tunggal. Nilai ini mewakili keluaran konvolusi pada posisi tersebut.

Setelah dilakukan perhitungan konvolusi maka dilakukan langkah *Stride*. *Filter* bergerak melintasi gambar dengan ukuran langkah tertentu yang dikenal dengan *stride* (Wani et al., 2021). Langkah tersebut menentukan seberapa banyak *filter* menggeser posisi pada setiap langkah. Langkah yang lebih besar mengurangi dimensi spasial dari keluaran *feature map*.

Convolutional Layer. Konvolusi berbentuk kernel 3x3 atau 5x5 yang akan bergeser ke seluruh matriks dengan operasi *dot product*. Rumus umum proses konvolusi sebagai berikut :

$$(I * K)(x, y) = \sum_{i=-2}^2 \sum_{j=-2}^2 I(x + i \cdot S, y + j \cdot S) \cdot K(i, j) \quad (1)$$

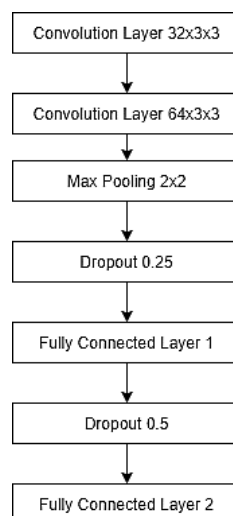
Keterangan :

- (I)(x,y) = Nilai piksel pada posisi (x,y) pada citra *input image*.
- K(i, j) = Nilai kernel atau *filter* pada posisi (i,j).
- (x, y) = Posisi piksel pada citra *output*.
- i dan j = Indeks yang mengiterasi elemen-elemen kernel.
- S = Langkah (*stride*), yang menentukan seberapa jauh langkah konvolusi
- m dan n = Parameter yang menentukan jangkauan indeks kernel

Berikutnya adalah langkah *padding* yang mengacu pada penambahan baris dan kolom tambahan nol (atau nilai lain) di sekitar gambar masukan. *Padding* dapat membantu mengontrol dimensi spasial dari peta fitur keluaran (Alrasheedi et al., 2023). Setelah *padding* selesai, langkah selanjutnya adalah *activation function*, sebuah fungsi aktivasi *ReLU* atau bisa juga fungsi lain seperti *sigmoid* atau *tanh* diterapkan untuk data yang bersifat non-linier.

Gambar 2 menunjukkan arsitektur Neural Network yang diusulkan. Ukuran masukannya adalah 224 x 224 x 1 dan gambar memiliki satu *channel* yaitu skala abu-abu. Lapisan konvolusi pertama menggunakan 32 kernel 3x3. Lapisan konvolusi kedua menggunakan 64 *kernel* 3x3. Setelah konvolusi akan dilanjutkan ke *layer max pooling* ukuran 2x2. Setelah menemukan nilai tertinggi menggunakan *max pooling*, dilanjutkan dengan *dropout*. Langkah ini dilakukan agar *dataset* pelatihan tidak mengalami *overfitting*. Proses terakhir *fully connected* dengan 5, karena *output* yang digunakan adalah 5 kelas.

Pada setiap layar konvolusi terdapat *hidden layer* dan *activation function ReLu*. Aktifasi *ReLU* berfungsi dapat juga mengurangi *overfitting* yang terjadi. *ReLU* mengubah semua nilai *negative* menjadi nol sehingga meningkatkan efisiensi kinerja model untuk



menangani informasi. Serta *ReLU* dapat mengatasi masalah *vanishing gradient* karena gradien dari *ReLU* adalah konstan 1 untuk nilai positif. *Vanishing gradient* bisa terjadi karena saat inialisasi bobot awal sangat kecil atau sangat besar.

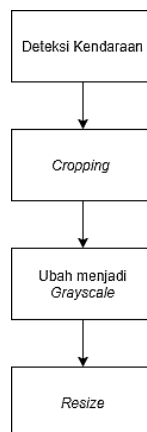
Gambar 2. Arsitektur CNN

2.3 Pre processing

Tahap *preprocessing* dilakukan untuk mendapatkan objek kendaraan agar mempermudah pada saat *training* CNN. Tahapan deteksi dilakukan antara lain deteksi kendaraan, *cropping*, dan merubah gambar ke dalam 1 channel warna *grayscale*. Alur *preprocessing* dapat dilihat pada gambar 3.

Gambar kendaraan dikumpulkan dari kelompok 1 hingga kelompok 5 dengan ukuran dan pola yang beragam dan representatif. Setiap kelompok akan diberikan label kelas yang sesuai untuk dijadikan data latih. Setelah *dataset* diberi label, format gambar standar yang digunakan adalah JPG. Setelah itu akan dilakukan normalisasi pada *dataset* dengan mengubah rentang nilai piksel dari 0 – 255 menjadi 1 saluran sehingga dapat membantu meningkatkan stabilitas pelatihan dan konvergensi model. kemudian dilakukan langkah *resize* menjadi 224 x 224 untuk kebutuhan *training* dan *testing* dengan ukuran urut tersebut.

Setelah itu dilakukan *Image Generator* dengan tujuan untuk membuat versi varian gambar pada *dataset*. Ini dapat mencakup perubahan seperti rotasi, pergeseran, *flip*, *zoom*, dan perubahan warna. dan juga menggunakan augmentasi data dapat membantu model untuk lebih memahami variasi dalam kumpulan data dan meningkatkan generalisasi. Pada *generator* gambar, *Image Augmented* digunakan di perpustakaan *keras* seperti *Sequential RandomFlip='horizontal'*, *RandomRotation=0,1*, *RandomZoom=0,1*, dan *Rescaling=1./255* (Kumar et al., 2022).



Gambar 3. Tahap *Preprocessing*

2.4 Model

CNN adalah kelas *deep neural networks* yang dirancang khusus untuk memproses data *grid* terstruktur, seperti gambar. Mereka terdiri dari beberapa lapisan, termasuk *convolutional layers*, *pooling layers*, dan *fully connected layers*. CNN sangat cocok untuk tugas-tugas seperti klasifikasi gambar karena mereka dapat secara otomatis mempelajari dan mengekstrak fitur yang relevan dari gambar, sehingga mengurangi kebutuhan rekayasa fitur manual.

ResNet (*Residual Network*) adalah arsitektur jaringan saraf dalam yang dikenal karena kemampuannya untuk melatih *deep neural network* secara efektif. ResNet50 adalah varian spesifik yang mencakup 50 lapisan, menggunakan *residual connections* (*skip connections*) untuk mengatasi masalah gradien yang hilang. Arsitektur ini telah menunjukkan performa luar biasa dalam berbagai tugas klasifikasi gambar, termasuk klasifikasi jenis kendaraan.

VGG (*Visual Geometry Group*) adalah arsitektur jaringan saraf konvolusional dalam lainnya yang menjadi runner-up dalam *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) pada tahun 2014. VGG16 adalah varian dari arsitektur VGG yang terdiri dari 16 lapisan bobot, termasuk 13 *convolutional layers* dan 3 *fully connected layers*. VGG16 dikenal karena kesederhanaan dan kemudahan pemahamannya (Naveen & Diwan, 2021).

Konfigurasi model mengacu pada pilihan dan pengaturan yang dibuat saat merancang dan menyiapkan arsitektur jaringan saraf. Konfigurasi model yang tepat sangat penting karena berdampak langsung pada performa, perilaku, dan efisiensi model pembelajaran mendalam.

2.4.1 Adam Optimizer

Adam optimizer digunakan pada model yang dilatih sehingga pengoptimal bertanggung jawab untuk mengelola bagaimana bobot model diperbarui selama pelatihan berdasarkan *loss function*. Adapun Rumus *adam optimizer* sebagai berikut :

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \quad (2)$$

Keterangan :

m_t = Nilai momentum pada iterasi ke- t

β_1 = Koefisien untuk momentum, sering kali memiliki nilai yang mendekati

m_{t-1} = Nilai momentum pada iterasi sebelumnya

g_t = Gradien dari *loss function* terhadap parameter yang dioptimalkan

2.4.2 Adaptive Moment Estimation

Adalah pengoptimal yang umum digunakan dan sering kali bekerja dengan baik secara empiris. Ini menggabungkan beberapa konsep dari optimasi SGD (*Stochastic Gradient Descent*) dengan momentum (Kumar et al., 2022). Langkah pembaruan pengoptimal Adam untuk setiap parameter θ pada langkah waktu t . Adapun rumus *adaptive moment estimation* sebagai berikut.

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot (g_t^2) \quad (3)$$

Keterangan :

v_t = Nilai momentum kuadrat gradien pada iterasi ke- t .

β_2 = Koefisien untuk momentum kuadrat gradien

v_{t-1} = Nilai momentum kuadrat gradien pada iterasi sebelumnya

g_t^2 = Kuadrat dari gradien pada iterasi ke- t .

2.4.3 Loss Function

Loss function digunakan dalam masalah klasifikasi *multiple class* dengan mengukur perbedaan antara probabilitas kelas yang diprediksi dan label kelas sebenarnya. Rumus perhitungan *loss function* sebagai berikut:

$$CCE = -\sum(y_i \cdot \log(\hat{y}^i)) \quad (4)$$

Keterangan :

- y_i = Nilai sebenarnya (*true value*) dari kelas ke-i.
- y^i = Probabilitas prediksi untuk kelas ke-iii yang dihasilkan oleh model.
- Σ = Simbol penjumlahan, yang menjumlahkan nilai untuk semua kelas.

Loss Function, juga dikenal sebagai a *cost function or objective function*, adalah fungsi matematika yang mengukur perbedaan antara keluaran yang diprediksi dan target sebenarnya untuk masukan tertentu dalam tugas *machine learning* (Yessou et al., 2020). Tujuan selama pelatihan adalah untuk meminimalkan *loss function* ini, karena fungsi ini sebagai ukuran seberapa baik performa model yang dilatih.

Evaluation metrics adalah metrik yang digunakan dalam *deep learning*, terutama untuk tugas klasifikasi. Tugasnya mengukur proporsi *instance* yang diklasifikasikan dengan benar dalam kumpulan data, memberikan cara mudah untuk menilai kinerja model secara keseluruhan (Hicham et al., 2018).

3. HASIL DAN PEMBAHASAN

Dataset yang digunakan dibagi menjadi 2 bagian sebelum dilakukan pelatihan, yaitu *dataset* pelatihan dan *dataset* pengujian untuk prediksi tipe grup. *Dataset* sebanyak 500 dibagi menjadi 80% sebagai pelatihan dan 20% sebagai pengujian. Proses pelatihan dan pengujian dilakukan menggunakan CNN, VGG16 dan Resnet50 dalam 75 epoch dengan. Skenario dilakukan dengan tujuan untuk mendapatkan nilai akurasi pada masing-masing metode.

3.1 Model CNN

Pada tabel 1 proses pelatihan model CNN pada epoch ke-75 dengan *batch* 13 per *epoch* membutuhkan waktu 9s 663ms/langkah untuk menyelesaikan setiap batch selama proses pelatihan. Hasil pengujian kinerja model CNN adalah *Training Loss* 0,099, *Training Accuracy* 0,9575, *Validation Loss* 0,52 dan *Validation Accuracy* 0,82. Hal ini menunjukkan Model CNN mencapai akurasi pelatihan yang tinggi sebesar 0,9575, bahwa model tersebut belajar mengklasifikasikan jenis kendaraan secara efektif pada kumpulan data pelatihan.

Training Loss 0,0999 menunjukkan bahwa prediksi model sebenarnya sangat cocok dengan jenis kendaraan dalam kumpulan data pelatihan. *Accuracy Validation* 0,82 mewakili kemampuan model untuk menggeneralisasi data baru yang belum terlihat. *Accuracy* 0,82 pada set

Tabel 1. Akurasi pada *epoch* ke 75

	<i>Epoch</i>	<i>batch/epoch</i>	<i>step</i>	<i>Loss (%)</i>	<i>Accuracy (%)</i>	<i>val_loss (%)</i>	<i>val_accuracy (%)</i>
CNN	75	13	9s 663ms	0,09	0,95	0,52	0,82
Resnet50	75	13	16s 1s	0,73	0,71	0,7	0,74
VGG16	75	13	29s 2s	0,14	0,93	0,25	0,91

Validasi merupakan tanda positif, yang menunjukkan bahwa model memiliki performa cukup baik pada data yang tidak ditemukan selama pelatihan. *Loss Validation* 0,5278 memberikan wawasan tentang performa generalisasi model. *Loss Validation* yang lebih rendah menunjukkan generalisasi yang lebih baik.

Secara keseluruhan, hasil yang dibagikan menunjukkan bahwa model CNN bekerja secara efektif dan menunjukkan kinerja yang wajar untuk tugas klasifikasi jenis kendaraan (Padilla et al., 2020). Model CNN menganalisis kesalahan klasifikasi, menyelidiki potensi sumber kesalahan, dan mempertimbangkan teknik interpretasi model untuk mendapatkan lebih banyak wawasan tentang perilaku model yang digunakan.

3.2 Model ResNet50

Pada tabel 1, proses pelatihan model ResNet50 pada *epoch* ke-75 dengan 13 *batch* per *epoch* membutuhkan waktu 16s 1s/langkah untuk menyelesaikan setiap *batch* selama proses pelatihan. Hasil pengujian kinerja model CNN adalah *Training Loss* 0,7317, *Training Accuracy* 0,71, *Validation Loss* 0,7074 dan *Validation Accuracy* 0,74. Hal ini menunjukkan Model ResNet50 belajar dari data pelatihan, namun mungkin masih ada ruang untuk perbaikan. Akurasi validasi sebesar 0,74 menunjukkan bahwa model melakukan generalisasi dengan cukup baik terhadap validasi data yang tidak terlihat, namun akurasi yang diperoleh ResNet50 pada *epoch* ke-75 tidak sebaik CNN dengan akurasi 0,9575.

Ukuran kumpulan data sebesar 500 relatif kecil untuk model ResNet50, sehingga melatih jaringan neural dalam secara efektif dapat menjadi tantangan tersendiri. Perbaikan yang berpotensi meningkatkan performa model seperti *Hyperparameter Tuning* dengan cara menentukan kecepatan pembelajaran, *batchsize*, dan jumlah *epoch* untuk menemukan pengaturan optimal pada kumpulan data serta Menerapkan teknik augmentasi data untuk meningkatkan ukuran dan keragaman kumpulan data pelatihan.

3.3 Model VGG16

Pada tabel 1, proses pelatihan model VGG16 pada *epoch* ke-75 dengan 13 *batch* per *epoch* membutuhkan waktu 29s 2s/langkah untuk menyelesaikan setiap *batch* selama proses pelatihan. Hasil pengujian kinerja model CNN adalah *Training Loss* 0,1482, *Training Accuracy* 0,9300, *Validation Loss* 0,2523 dan *Validation Accuracy* 0,9100.

Model VGG16 tampaknya bekerja dengan baik dalam tugas klasifikasi jenis kendaraan. Akurasi pelatihan sebesar 93,00% menunjukkan bahwa model telah belajar secara efektif dari data pelatihan, dan *Training Loss* sebesar 0,1482 menunjukkan bahwa model meminimalkan kesalahan prediksi. Akurasi VGG16 pada *epoch* ke-75 menunjukkan hasil sebesar 0,93, sehingga hampir menyamai akurasi CNN yang sebesar 0,9 akurasi VGG16 terbaik dari CNN dan ResNet50 untuk mengenali jenis kendaraan.

Akurasi validasi sebesar 91,00% dan kerugian validasi sebesar 0,2523 menunjukkan bahwa model dapat menggeneralisasi dengan baik terhadap data yang tidak terlihat, karena model ini mempertahankan akurasi yang tinggi pada kumpulan data validasi.

Ukuran kumpulan data sebanyak 500 sampel relatif kecil dibandingkan model VGG16, namun model mencapai hasil yang baik, yang menunjukkan efektivitas VGG16 sebagai *classifier* jenis kendaraan. Untuk meningkatkan performa model VGG16 tentunya terdapat banyak variasi teknik yaitu *hyperparameter*, *data augmented*, atau penggunaan *dataset* yang lebih besar jika tersedia. Selain itu, penting untuk mempertimbangkan potensi *overfitting* dan mempertimbangkan *trade-off* antara kompleksitas model dan ukuran kumpulan data.

3.4 Model Validation

Gambar 4 menunjukkan grafik *train* dan *validation loss* sebagai model terbaik. Grafik *training loss* sebagai model terbaik karena menunjukkan tren penurunan yang stabil seiring dengan meningkatnya akurasi selama proses pelatihan. Hal ini menunjukkan bahwa model yang diusulkan secara konsisten mempelajari pola dalam data pelatihan dan meminimalkan kesalahan prediksi. grafik *loss validation* juga menunjukkan tren penurunan yang stabil atau konsisten selama pelatihan pada data pelatihan. Hal ini menunjukkan bahwa model dapat menggeneralisasi dengan *data validation* yang baik, yang belum pernah di *training* sebelumnya.

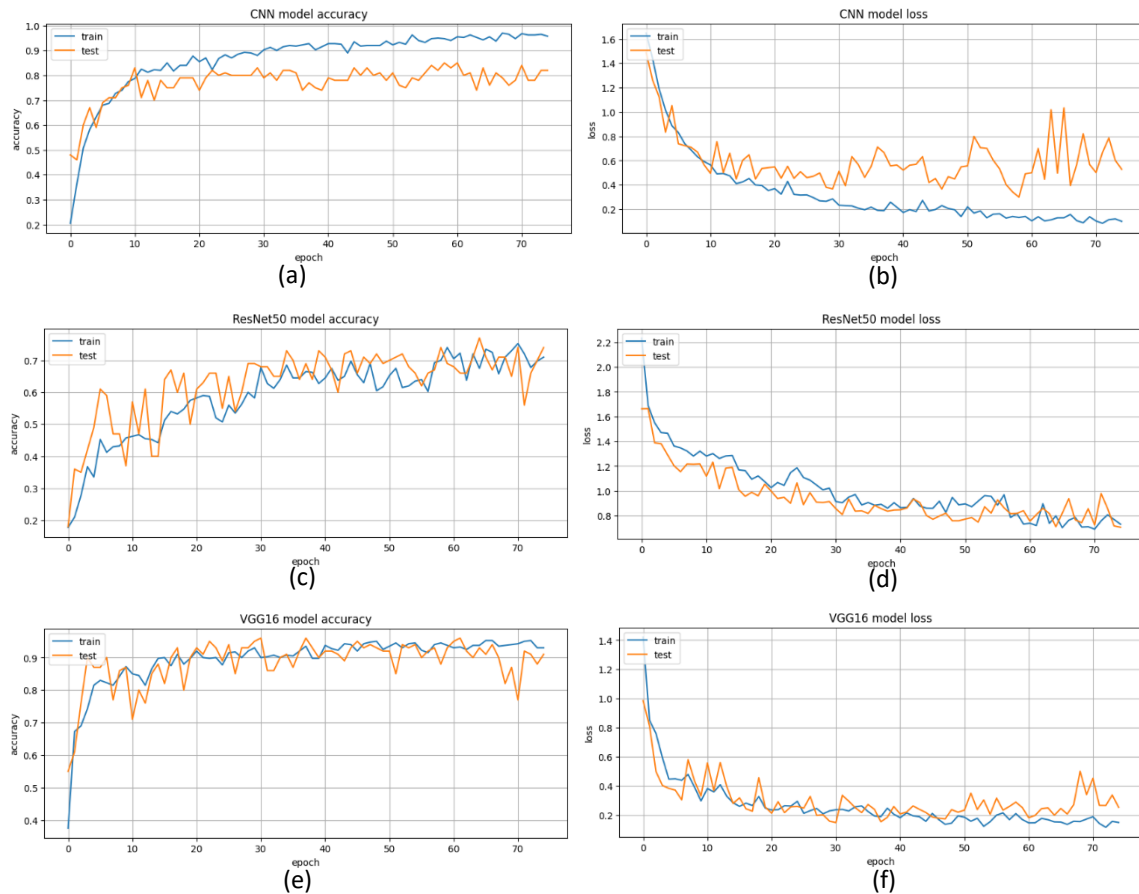
Di CNN, grafik *loss validation* mengalami penurunan drastis setelah *epoch* 6, namun tetap konstan mendekati grafik *train loss*. Perbedaan kecil antara grafik *train loss* dan grafik *validation loss* menunjukkan bahwa model tidak mengalami *overfitting* pada data pelatihan dan dapat melakukan generalisasi dengan baik pada data baru. Grafik yang ideal akan menunjukkan tren penurunan yang stabil baik untuk *training loss* maupun *loss value*, dengan perbedaan nilai yang kecil di antara keduanya. Hal ini menunjukkan bahwa model telah belajar secara efektif, tidak *overfitting*, dan dapat melakukan generalisasi dengan baik pada data baru.

Akurasi model CNN, ResNet50, VGG16 dapat dilihat pada tabel 2. Dimana akurasi pelatihan VGG16 mendapatkan hasil tertinggi dengan 0.91 antara CNN 0.82 dan ResNet50 0.74, serta *validation loss* sebesar 0.25 antara ResNet50 0.7 dan CNN 0.52. saat *validation loss*, VGG16 mendapat hasil terkecil antara CNN 0.52 dan ResNet16 0.7, serta *Validation Accuracy* VGG16 mendapat nilai lebih tinggi dibandingkan CNN 0.82 dan ResNet 0.74. Jadi pada penelitian ini dengan objek berupa bentuk atau pola kendaraan CNN mendapat hasil 0,95, disusul VGG16 dengan hasil 0,93 dan ResNet50 dengan hasil 0,71.

Tabel 2. Akurasi pada 3 model

	<i>Loss</i> (%)	<i>Accuracy</i> (%)	<i>Validation Loss</i> (%)	<i>Validation Accuracy</i> (%)
CNN	0.09	0,95	0,52	0,82
Resnet50	0,73	0,71	0,7	0,74
VGG16	0,14	0,93	0,25	0,91

3.5 Best Prediction Model



Gambar 4. Train and Validation Loss Model Validation.
(a) CNN Accuracy, (b) CNN Validation Loss, (c) ResNet50 Accuracy,
(d) ResNet50 Validation Loss, (e) VGG16 Accuracy, (f) VGG16 Validation Loss

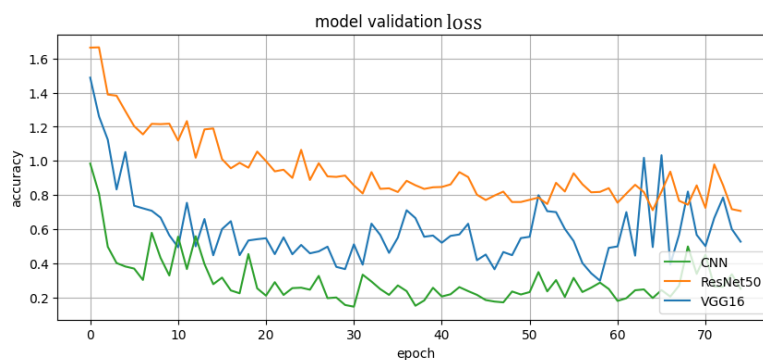
Gambar 4a menampilkan akurasi model *training* CNN lebih tinggi dari akurasi *test* pada *epoch* 75. Gambar 4b menunjukkan model *loss training* CNN lebih rendah dari model *loss test* pada *epoch* 75, sehingga untuk model CNN lebih baik akurasi dan *loss* pada saat *training* untuk objek kendaraan. Gambar 4c menampilkan akurasi model *test* ResNet50 lebih tinggi dari akurasi *train* pada *epoch* 75. Gambar 4d menunjukkan model *loss test* ResNet50 lebih rendah dari model *loss train* pada *epoch* 75, sehingga untuk model ResNet50 lebih baik akurasi dan *loss* pada saat *test* untuk objek kendaraan. Gambar 4e menampilkan akurasi model *training* VGG16 lebih tinggi dari akurasi *test* pada *epoch* 75. Gambar 4f menunjukkan model *loss training* VGG16 lebih rendah dari model *loss test* pada *epoch* 75, sehingga untuk model VGG16 lebih baik akurasi dan *loss* pada saat *training* untuk objek kendaraan.

Pada gambar 4a, 4c dan 4e model usulan CNN, ResNet50 dan VGG16, grafik *model validation accuracy* menunjukkan bahwa VGG16 dari *epoch* 1 memiliki akurasi yang lebih tinggi dibandingkan CNN dan ResNet50, pada *validation loss* grafik menunjukkan bahwa dari *epoch* 1 CNN mendapatkan *error* yang lebih kecil dibandingkan VGG16 dan ResNet50. VGG16 memiliki peningkatan *validation accuracy* yang lebih stabil dibandingkan CNN dan ResNet50, untuk *val loss*, penurunan stabil pada grafik ditunjukkan oleh ResNet50.

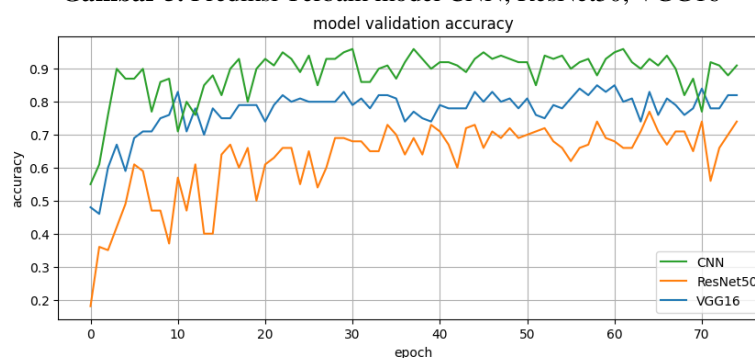
Kedalaman dan kompleksitas arsitektur VGG16 memainkan peran penting dalam kemampuannya menangkap dan mempelajari fitur dari gambar 4 atau data spasial

lainnya. Jaringan yang lebih dalam mempunyai potensi untuk mempelajari fitur yang lebih kompleks dan abstrak (Aydin et al., 2022), tetapi mungkin juga lebih mahal secara komputasi dan lebih sulit untuk dilatih. terdiri dari 16 lapisan konvolusional dan terhubung sepenuhnya. Hal ini dapat mempermudah pelatihan pada kumpulan data yang lebih kecil atau dengan sumber daya komputasi yang terbatas. Namun, hal ini mungkin lebih rentan terhadap *overfitting* pada kumpulan data yang sangat besar.

ResNet50, bagian dari ResNet50 (Jaringan Residu), lebih dalam dan kompleks. menggunakan *skip connection* untuk mengurangi masalah *gradient loss*, membuatnya lebih mudah untuk melatih jaringan yang sangat dalam (Nagpal et al., 2022). Hal ini membuatnya lebih cocok untuk kumpulan data yang sangat besar dan tugas yang memerlukan ekstraksi fitur kompleks. Arsitektur jaringan CNN lebih sederhana daripada ResNet50 dan VGG16 dan VGG16 dan ResNet50 memiliki kemampuan untuk melatih data yang besar (Chilakalapudi et al., 2022) namun dalam evaluasinya model CNN mendapatkan *overfitting* karena *gap* antara *validation accuracy* dan *accuracy* sebesar 0.13%. mungkin inilah alasan mengapa VGG16 mendapatkan akurasi lebih tinggi dibandingkan ResNet50 dan CNN. Gambar 5 menunjukkan prediksi jenis kendaraan dengan 3 model. Pada validasi *training* di *epoch 75* grafik menunjukkan validasi CNN lebih tinggi dari ResNet50 dan VGG16.



Gambar 5. Prediksi Terbaik model CNN, ResNet50, VGG16



Gambar 6. Validation Loss untuk model CNN, ResNet50 and VGG16

Gambar 6 menunjukkan grafik CNN lebih rendah dari ResNet50 dan VGG16 untuk validasi *loss training* di *epoch 75*. Banyak factor yang menentukan CNN lebih tinggi dari 2 metode yang lain. *Factor* yang terjadi dikarenakan CNN dapat melakukan *training* pada *dataset* yang sedikit.

4. KESIMPULAN

1. VGG16 bisa lebih baik daripada *Convolutional Neural Network* (CNN) atau ResNet50 untuk klasifikasi jenis kendaraan bergantung pada berbagai faktor, termasuk kumpulan data, kompleksitas tugas, dan sumber daya komputasi yang tersedia.
2. *Domain-Specific Features*, untuk klasifikasi jenis kendaraan sering kali melibatkan pengenalan fitur spesifik seperti bentuk mobil, lampu depan dan lampu belakang. Kemudian dapat membuat VGG16 khusus yang dirancang dengan arsitektur jaringan yang mempertimbangkan *Domain-Specific Features*.
3. *Model Size*, model pelatihan seperti VGG16 dan ResNet50 memiliki jumlah parameter yang relatif besar. Dalam beberapa kasus, CNN khusus dirancang agar lebih ringkas dengan tetap menjaga atau bahkan meningkatkan akurasi klasifikasi. Hal ini sangat penting untuk penerapan pada perangkat atau aplikasi dengan sumber daya terbatas yang mengutamakan ukuran model.
4. *Mitigating Overfitting*, tugas klasifikasi jenis kendaraan mungkin memerlukan perbedaan yang sangat jelas antar jenis kendaraan, yang dapat menyebabkan *overfitting* saat menggunakan model berukuran besar. Dengan merancang CNN khusus, VGG16, dan ResNet50, kompleksitas model dapat dikontrol dan menggunakan teknik seperti *regularisasi* dan *dropout* untuk mengurangi *overfitting* pada kumpulan data tertentu.
5. *Transfer Learning*, meskipun VGG16 dan ResNet50 dapat disesuaikan untuk tugas tertentu, melatih CNN atau VGG16 khusus dari awal memungkinkan inisialisasi jaringan dengan bobot acak. Pada klasifikasi jenis kendaraan, hal ini dapat menghasilkan model yang lebih disesuaikan dengan tugas klasifikasi jenis kendaraan.

5. SARAN

Arsitektur VGG16 perlu disesuaikan agar berfokus pada karakteristik kendaraan yang paling relevan, sehingga berpotensi menghasilkan performa yang lebih baik. klasifikasi jenis kendaraan mungkin memerlukan perbedaan yang sangat jelas antar jenis kendaraan untuk menghindari *overfitting* serta dapat menerapkan Teknik *dropout* atau *regularization* pada *dataset* yang dilatih. Lalu perlu menentukan bobot dengan acak pada saat inisialisasi jaringan dengan menyesuaikan tugas untuk klasifikasi kendaraan.

UCAPAN TERIMA KASIH

Ucapan terima kasih diberikan kepada semua pihak yang telah membantu dalam penyelesaian perancangan pengenalan jenis golongan kendaraan, serta seluruh bimbingan, arahan, dan dukungan yang telah diberikan selama perancangan ini berlangsung.

DAFTAR PUSTAKA

- Alrasheedi, F., Zhong, X., & Huang, P. C. (2023). Padding Module: Learning the Padding in Deep Neural Networks. *IEEE Access*.
<https://doi.org/10.1109/ACCESS.2023.3238315>
Aug, T. I. P., Jan, T. I. P., Jan, T. I. P., Costa, D., Syn-, T., Jan, T. I. P., Jan, T. I. P., April,

- T. I. P., Salah, A., June, T. I. P., & June, T. I. P. (2015). 2015 Index IEEE Transactions on Image Processing Vol. 24. *IEEE Transactions on Image Processing*, 24(12), 6092–6143. <https://doi.org/10.1109/tip.2015.2506438>
- Aydin, I., Budak, G., Sefer, A., & Yapar, A. (2022). CNN-Based Deep Learning Architecture for Electromagnetic Imaging of Rough Surface Profiles. *IEEE Transactions on Antennas and Propagation*. <https://doi.org/10.1109/TAP.2022.3177493>
- Chen, Y., Zhu, W., Yao, D., & Zhang, L. (2017). Vehicle type classification based on convolutional neural network. *Proceedings - 2017 Chinese Automation Congress, CAC 2017*. <https://doi.org/10.1109/CAC.2017.8243078>
- Chilakalapudi, H. P., Venkatesan, R., & Kamatham, Y. (2022). Architecture-based Evaluation of VGG16 and ResNet Models for an Online Deep Learning Environment for Medical Applications. *ICISTSD 2022 - 3rd International Conference on Innovations in Science and Technology for Sustainable Development*. <https://doi.org/10.1109/ICISTSD55159.2022.10010588>
- Hicham, B., Ahmed, A., & Mohammed, M. (2018). Vehicle Type Classification Using Convolutional Neural Network. *Colloquium in Information Science and Technology, CIST*. <https://doi.org/10.1109/CIST.2018.8596500>
- Huang, C., Liang, B., Li, W., & Han, S. (2017). A convolutional neural network architecture for vehicle logo recognition. *Proceedings of 2017 IEEE International Conference on Unmanned Systems, ICUS 2017*. <https://doi.org/10.1109/ICUS.2017.8278355>
- Khagi, B., & Kwon, G. R. (2021). Convolutional Neural Network-Based Natural Image and MRI Classification Using Gaussian Activated Parametric (GAP) Layer. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2021.3093455>
- Kumar, G. P., Priya, G. S., Dileep, M., Raju, B. E., Shaik, A. R., & Sarman, K. V. S. H. G. (2022). Image Deconvolution using Deep Learning-based Adam Optimizer. *6th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2022 - Proceedings*. <https://doi.org/10.1109/ICECA55336.2022.10009073>
- Le Callet, P., Viard-Gaudin, C., & Barba, D. (2006). A convolutional neural network approach for objective video quality assessment. *IEEE Transactions on Neural Networks*. <https://doi.org/10.1109/TNN.2006.879766>
- Nagpal, P., Bhinge, S. A., & Shitole, A. (2022). A Comparative Analysis of ResNet Architectures. *2022 International Conference on Smart Generation Computing, Communication and Networking, SMART GENCON 2022*. <https://doi.org/10.1109/SMARTGENCON56628.2022.10083966>
- Naveen, P., & Diwan, B. (2021). Pre-trained VGG-16 with CNN architecture to classify X-Rays images into normal or pneumonia. *2021 International Conference on Emerging Smart Computing and Informatics, ESCI 2021*. <https://doi.org/10.1109/ESCI50559.2021.9396997>
- Padilla, R., Netto, S. L., & Da Silva, E. A. B. (2020). A Survey on Performance Metrics for Object-Detection Algorithms. *International Conference on Systems, Signals, and Image Processing*. <https://doi.org/10.1109/IWSSIP48289.2020.9145130>
- Pang, Y., Sun, M., Jiang, X., & Li, X. (2018). Convolution in convolution for network in network. *IEEE Transactions on Neural Networks and Learning Systems*. <https://doi.org/10.1109/TNNLS.2017.2676130>
- Pramana, A. L., Setyati, E., & Kristian, Y. (2020). MODEL CNN LENET DALAM

PENGENALAN JENIS GOLONGAN KENDARAAN PADA JALAN TOL. *Jurnal Teknik*. <https://doi.org/10.30736/jt.v13i2.469>

- Wani, T. M., Gunawan, T. S., Qadri, S. A. A., Mansor, H., Arifin, F., & Ahmad, Y. A. (2021). Stride Based Convolutional Neural Network for Speech Emotion Recognition. *2021 IEEE 7th International Conference on Smart Instrumentation, Measurement and Applications, ICSIMA 2021*. <https://doi.org/10.1109/ICSIMA50015.2021.9526320>
- Yessou, H., Sumbul, G., & Demir, B. (2020). A Comparative Study of Deep Learning Loss Functions for Multi-Label Remote Sensing Image Classification. *International Geoscience and Remote Sensing Symposium (IGARSS)*. <https://doi.org/10.1109/IGARSS39084.2020.9323583>