Pengenalan Golongan Jenis Kendaraan Bermotor pada Ruas Jalan Tol Menggunakan CNN

Ricky Herwanto, Kartika Gunadi, Endang Setyati
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) – 8417658
E-Mail: rickyherwanto75@gmail.com, kgunadi@petra.ac.id, endang@stts.edu

ABSTRAK

Sistem pembayaran di gerbang tol sudah mengalami kemajuan, yaitu dari pembayaran menggunakan uang fisik digantikan dengan *e-money*. Sistem pembayaran perlu mengetahui golongan jenis kendaraan yang masuk pada gerbang tol sehingga sistem bisa tahu berapa biaya yang harus diambil dari saldo *e-money*. Terdapat lima golongan jenis kendaraan, tetapi masih banyak gerbang tol yang memiliki batas tinggi untuk membatasi golongan kendaraan yang bisa masuk sehingga menyulitkan golongan selain golongan I karena hanya memiliki gerbang yang sedikit. Dengan berkembangnya teknologi, golongan kendaraan dapat dikenali secara otomatis menggunakan *Neural Network*.

Penelitian ini menggunakan metode *You Only Look Once* dan *Convolutional Neural Network*. *You Only Look Once* digunakan untuk mendeteksi posisi kendaraan pada gambar. *Convolutional Neural Network* digunakan untuk mengklasifikasikan golongan jenis kendaraan pada gambar. Untuk model *convolutional neural network*, salah satu model yang terkenal adalah VGG16 yang cukup baik dalam mengklasifikasikan gambar.

Hasil penelitian yang akan ditampilkan adalah hasil pengenalan golongan jenis kendaraan dalam bentuk *string*. Hasil dari pengujian yang dilakukan adalah akurasi dari konfigurasi *convolutional neural network* yang digunakan adalah 93.5% dan *f-score* sebesar 81.37% dan model VGG16 dengan akurasi sebesar 90.76% dan *f-score* sebesar 73.53%

Kata Kunci: *Convolutional Neural Network, VGG16, Golongan Kendaraan, You Only Look Once*

ABSTRACT

Payment system at the toll gate has been improved, from using physical money replace with e-money. The system needs to which class types of the vehicle entering the toll gate so the system can know how much will it take from the e-money. There are five class types of vehicles, but there are still many toll gates that have high limit to limit the class of vehicles that can enter, making it difficult for class types other than the first class type because they only have a few gates.

This research uses You Only Look Once and Convolutional Neural Network as its methods. You Only Look Once is used to detect the location of the vehicle in the image. Convolutional Neural Network is used to classify the class types of the vehicle in the image. For convolutional neural network model, one well-known model is VGG16 which is good in classifying images.

The result of this research that will be displayed is the classified of the class type of the vehicle in the form of strings. The result from tests that were done is an accuracy of 93.5% and f-score of 81.37% from self-configuration convolutional neural network and an accuracy of 90.76% and f-score of 73.53% for VGG16 model.

Keywords: *Convolutional Neural Network, VGG16, Class Types of Vehicles, You Only Look Once*

1. PENDAHULUAN

Sistem pembayaran di gerbang tol saat ini di Indonesia sudah mengalami kemajuan, yaitu dari pembayaran menggunakan uang fisik digantikan dengan *e–money*. Sistem pembayaran di gerbang tol tersebut perlu mengetahui golongan jenis kendaraan yang masuk pada gerbang tol, sehingga bisa tahu berapa biaya yang harus diambil dari saldo *e-money*.

Terdapat lima golongan jenis kendaraan bermotor, masing-masing jenis memiliki kriterianya tersendiri. Kebanyakan dari gerbang tol saat ini memiliki batas tinggi karena gerbang tol tersebut dikhususkan untuk kendaraan jenis mobil penumpang seperti sedan yang merupakan golongan jenis kendaraan golongan I. Hal ini menyebabkan golongan lain hanya memiliki sedikit gerbang, bahkan biasanya hanya satu gerbang untuk golongan selain golongan I.

Untuk dapat menghasilkan sistem pengenalan golongan jenis kendaraan, dibutuhkan pendekatan secara komputerisasi dengan menggunakan metode *you only look once* (YOLO) dan *convolutional neural network* (CNN). Salah satu arsitektur CNN yang digunakan adalah VGG16, dimana *convolutional neural network* model ini secara signifikan lebih unggul dibandingkan model-model sebelumnya. Kedua metode ini dinilai sebagai metode dengan performa terbaik pada saat ini untuk pengenalan objek dan klasifikasi gambar.

Sistem pengenalan golongan jenis kendaraan ini menerima input berupa gambar kendaraan, lalu setelah itu melakukan pencarian posisi atau letak kendaraan pada gambar menggunakan YOLO, lalu letak kendaraan pada gambar tersebut akan dibuatkan kotak (*bounding box*). Pengenalan golongan jenis kendaraan akan dilakukan menggunakan CNN pada kotak yang telah dibuat. Dengan membatasi posisi kendaraan dengan kotak, akurasi yang dihasilkan juga bisa lebih tinggi.

2. DASAR TEORI

2.1 Golongan Jenis Kendaraan

Diambil dari Keputusan Menteri Pekerjaan Umum Nomor 370/KPTS/M/2007 [2] dijelaskan bahwa golongan jenis kendaraan dibagi menjadi 5 golongan. Daftar golongan jenis kendaran dapat dilihat pada Tabel 1.

Tabel 1. Golongan Kendaraan pada Jalan Tol

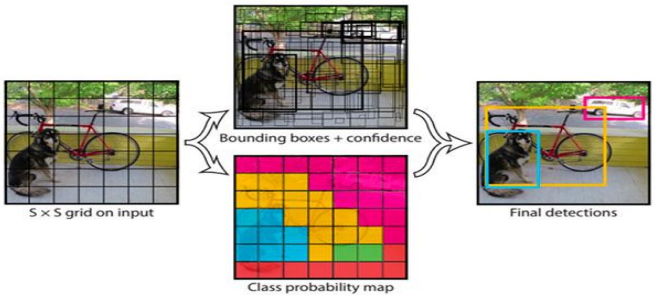
Golongan	Jenis Kendaraan
Golongan I	Sedan, Jip, Pick Up/Truk Kecil, dan Bus
Golongan II	Truk dengan 2 (dua) gandar
Golongan III	Truk dengan 3 (tiga) gandar
Golongan IV	Truk dengan 4 (empat) gandar
Golongan V	Truk dengan 5 (lima) gandar atau lebih

2.2 *You Only Look Once*

You Only Look Once (YOLO) merupakan jaringan untuk mendeteksi objek. Mendeteksi objek terdiri dari menentukan lokasi pada gambar dimana terdapat objek tertentu. YOLO menerapkan *single neural network* pada gambar dan akan membagi gambar menjadi grid berukuran SxS.

Pada setiap objek yang ada pada gambar, satu sel *grid* bertanggung jawab untuk memprediksi objek tersebut. Setiap sel *grid* memprediksi B *bounding boxes* dan juga C probabilitas kelas yang jumlahnya sudah pasti dan setiap *box* memiliki tingkat keyakinan. Setiap *bounding box* mempunyai 5 komponen yaitu koordinat x, koordinat y, lebar (*width*), tinggi (*height*), dan *confidence score* (nilai probabilitas *bounding box* yang bersangkutan memiliki sebuah objek). *Confidence score* merefleksikan seberapa besar kotak tersebut mengandung sebuah objek dan seberapa akurat *bounding box* tersebut. X dan y merupakan koordinat yang merepresentasikan titik tengah dari kotak, relatif terhadap lokasi sel *grid.* Koordinat ini akan dinormalisasi menjadikan nilainya dari 0 hingga 1. Lalu lebar dan tinggi dimensi kotak juga dinormalisasi menjadi 0 hingga 1, relatif terhadap ukuran gambar. Setiap sel mempunyai 20 kemungkinan kelas dimana objek yang terdeteksi dimiliki hanya pada satu kelas tertentu [3].

Deteksi menggunakan YOLO ini memiliki 24 *convolutional layer* yang dilanjutkan dengan 2 *fully connected layers*. Beberapa *convolutional layers* menggunakan ukuran *layer* 1 x 1 untuk mereduksi fitur-fitur dari *layer* sebelumnya. *Convolutional layer* dari YOLO sudah dilatih terlebih dahulu pada *ImageNet classification task*. Hasil *output* terakhir dari *network* YOLO adalah 7 x 7 x 30 prediksi tensor [5]. Diagram Algoritma YOLO dapat dilihat pada Gambar 1.



Gambar 1. Diagram Algoritma YOLO [5]

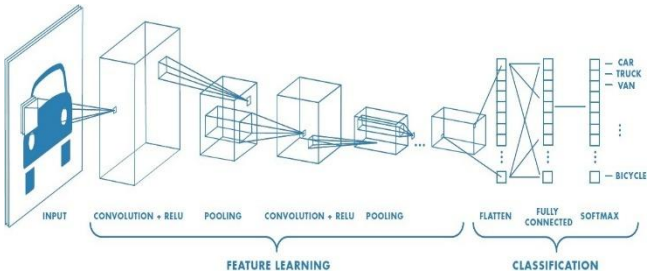
2.3 *Convolutional Neural Network*

Convolutional Neural Network (CNN) adalah bagian dari *neural network* yang berspesialisasi dalam memproses data yang memiliki topologi *grid*, seperti gambar. Gambar mengandung serangkaian *pixel* yang tersusun dalam bentuk *grid* yang mengandung nilai *pixel* yang menunjukkan seberapa terang dan warna dari *pixel* tersebut.

CNN pada umumnya memiliki tiga *layer*, yaitu *convolutional layer*, *pooling layer*, dan *fully connected layer*. *Convolutional layer* adalah inti dari arsitektur CNN. Pada *layer* ini dilakukan proses konvolusi *filter* berupa matriks berukuran MXN (pada umumnya adalah 3x3) untuk mengekstrak fitur yang menonjol pada gambar. Fitur yang diambil ini mengandung hubungan yang khusus pada setiap *pixel*. Konvolusi dari gambar dengan konvolusi *filter* yang berbeda dapat melakukan performa seperti pendeteksian tepi, *blur*, dan juga penajaman gambar. Dalam *convolutional layer* ini selain memilih *filter*, juga ada *stride*. *Stride* adalah jumlah perpindahan *pixel* pada matriks *input*. Ketika *stride* bernilai 1 maka perpindahan *filter* yang dilakukan juga 1 *pixel* setiap waktu. Setelah *filter* dan *stride* ada juga *padding*. Kadang *filter* yang diaplikasikan pada gambar *input* tidak sepenuhnya tepat, pada saat inilah *padding* dapat digunakan [4].

Pada *convolutional layer* ini juga dilakukan *activation function* pada setiap nilai dari *feature map*. ReLU *function* adalah *activation function* yang banyak digunakan di *neural network*. Tujuannya adalah mengubah semua input negatif menjadi nol [8].

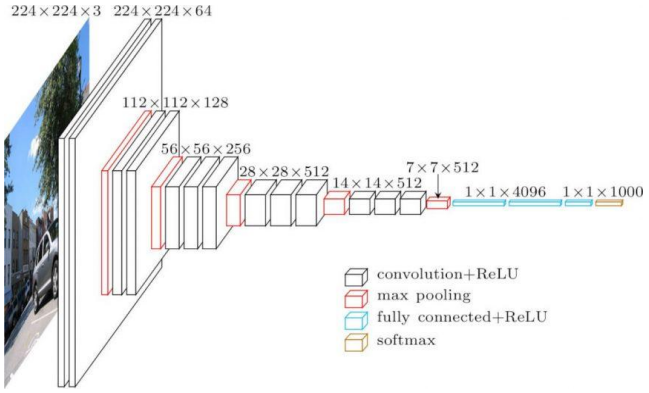
Setelah itu, pada *pooling layer* akan dilakukan pengecilan matriks, beberapa orang menyebutnya *reduce*. Dari setiap fitur yang dihasilkan dari *convolutional layer*, akan diambil nilai tertinggi pada setiap *window* untuk mengecilkan matriks [1]. Untuk arsitektur dari CNN dapat dilihat pada Gambar 2.



Gambar 2. Arsitektur CNN [4]

2.4 VGG16

VGG16 adalah sebuah model *convolutional neural network*. Model ini mencapai akurasi 92.7% *top*-5 di *ImageNet*. Model ini melakukan penyempurnaan dari model sebelumnya yaitu AlexNet dengan menggantikan fitur ukuran kernel yang besar dengan filter ukuran kernel 3x3 satu dengan lainnya. *Input* pada *convolutional layer* pertama adalah ukuran pasti yaitu 224 x 224 gambar RGB. Gambar akan melewati tumpukan dari *convolutional layer*, dimana *filter* digunakan dengan bidang reseptif yang sangat kecil yaitu 3x3. Pada salah satu konfigurasi, *convolution filters* 1x1 juga dimanfaatkan, yang bisa dilihat sebagai transformasi linier dari *channel input*. *Convolution stride* di pastikan adalah 1 *pixel*. *Spatial pooling* dilakukan oleh lima *max-pooling layer*. Lalu ada tiga *fully-connected layer* yang memiliki konfigurasi sama pada semua jaringan. Dan pada *layer* terakhir adalah *soft-max layer*. Setiap *hidden layer* dilengkapi dengan ReLU [7]. Arsitektur dari VGG16 dapat dilihat pada Gambar 3.

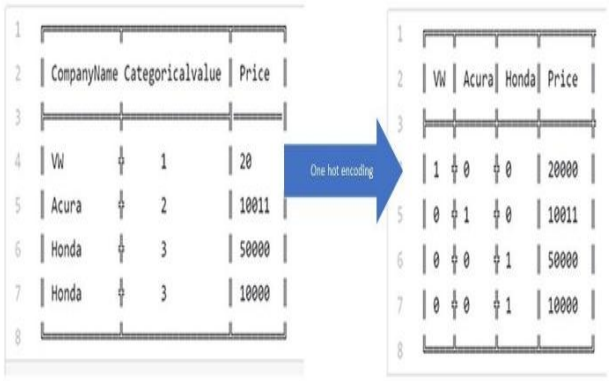


Gambar 3. Arsitektur VGG16 [10]

2.5 *One Hot Encoding*

One hot encoding adalah proses dimana kategori diubah menjadi bentuk yang dapat disediakan pada *machine learning* untuk dapat

memprediksi dengan lebih baik [9]. Untuk ilustrasi dari *one hot encoding* dapat dilihat pada Gambar 4.



Gambar 4. Ilustrasi *One Hot Encoding*

2.6 *Confusion Matrix*

Confusion matrix merupakan *table* yang menjelaskan performa dari sebuah model klasifikasi. *Confusion matrix* ini berisi informasi tentang kelas yang seharusnya dan hasil klasifikasi yang dilakukan oleh model yang telah dibuat, dan informasi ini digunakan untuk mengukur performa dari model tersebut. Ada dua tipe *confusion matrix* yaitu 2-*class confusion matrix* dan *multi-class confusion matrix*. Untuk mengukur performa akan dilakukan perhitungan dari isi tabel *confusion matrix [11]*.

Isi dari tabel *confusion matrix* ada 4, yaitu :

- *True Positive*, kondisi dimana kelas yang sebenarnya adalah benar dan hasil prediksi juga mengatakan benar.
- *False Negative*, kondisi dimana kelas yang sebenarnya seharusnya adalah benar tapi hasil prediksi mengatakan salah.
- *True Negative*, kondisi dimana kelas yang sebenarnya adalah salah dan hasil prediksi juga mengatakan salah.
- *False Positive*, kondisi dimana kelas yang sebenarnya seharusnya salah, tetapi hasil prediksi mengatakan benar.

Persamaan yang akan digunakan berdasarkan data dari *confusion matrix* adalah akurasi, *presisi*, *recall* dan *f-score*. Akurasi merupakan pengukuran seberapa benar sebuah sistem dapat mengklasifikasi dari keseluruhan. Akurasi dapat dihitung menggunakan persamaan (1).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

Precision merupakan rasio dari *true positive* dengan jumlah *true positive* dan *false positive*. *Precision* yang tinggi menunjukkan contoh yang di-*label* positif adalah memang positif. *Precision* dpaat dihitung dengan persamaan (2).

$$Precision = \frac{TP}{FP + TP} \quad (2)$$

Recall merupakan rasio dari *true positive* dengan jumlah *true positive* dan *false negative*. *Recall* yang tinggi menunjukkan bahwa kelas berhasil dikenali. *Recall* dapat dihitung dengan persamaan (3).

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Selain akurasi, *precision* dan *recall*, *confusion matrix* juga dapat digunakan untuk menghitung *F-Score* yang bertujuan untuk menghitung kombinasi dari *precision* dan *recall*. *F-Score* akan menggunakan *harmonic mean dari precision* dan *recall*. *F-Score* dapat dihitung dengan persamaan (4).

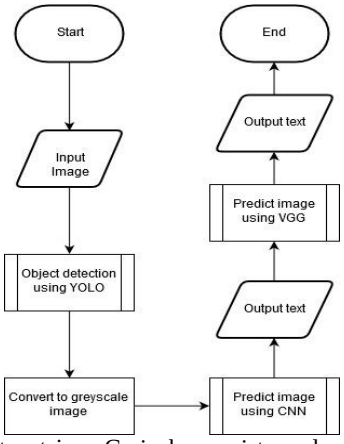
$$F\text{-}Score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (4)$$

3. DESAIN SISTEM

3.1 Analisis Sistem

Sistem pertama-tama akan menerima *input* data dari *user* berupa *image* kendaraan bermotor dalam format .jpg. Pada *image* tersebut kemudian akan diterapkan sistem untuk mendeteksi lokasi kendaraan bermotor yang ada di dalamnya. Sistem untuk mendeteksi lokasi kendaraan bermotor ini dilakukan menggunakan metode YOLO (*You Only Look Once*).

Setelah didapatkan gambar yang objeknya sudah di prediksi menggunakan YOLO, gambar tersebut akan dimasukkan ke dalam proses selanjutnya yaitu konversi gambar ke *grayscale*. Hasil gambar yang telah di *grayscale* akan diolah dan diproses menggunakan CNN (*Convolutional Neural Network)* yang telah di-train sebelumnya agar dapat menghasilkan jenis kendaraan dalam bentuk tipe data *string*. Setelah selesai dari proses CNN, maka gambar yang sama yang digunakan pada proses CNN akan digunakan kembali untuk diolah dan diproses menggunakan model VGG (*Visual Geometry Group*) yang telah di-train sebelumnya agar dapat menghasilkan jenis kendaraan dalam bentuk tipe data *string*. Garis besar sistem dapat dilihat pada Gambar 5.

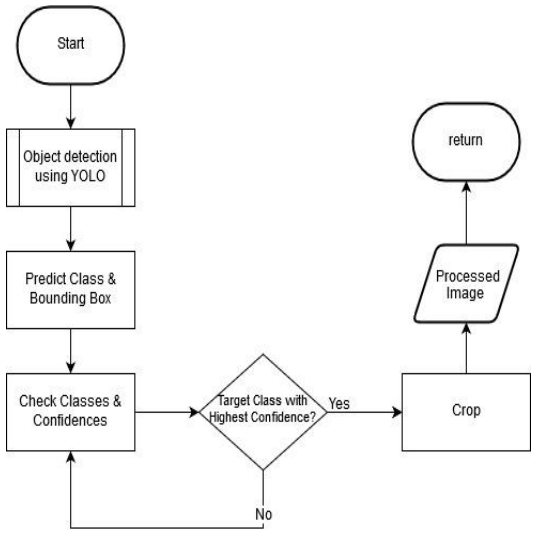


Gambar 5. Garis Besar Sistem

3.1.1. You Only Look Once

Sub-proses *You Only Look Once* digunakan untuk mendeteksi objek pada gambar. Pada sub-proses ini, objek yang akan dideteksi adalah kendaraan bermotor yang terdapat pada gambar. Kendaraan bermotor yang dideteksi adalah mobil, bis dan truk. Langkah pertama untuk proses ini adalah memprediksi semua kelas dan *bounding box* pada gambar, dimana pada proses ini semua objek akan dideteksi. Setelah itu akan dilakukan proses pengecekan *confidence*, dimana pada gambar dicari objek yang

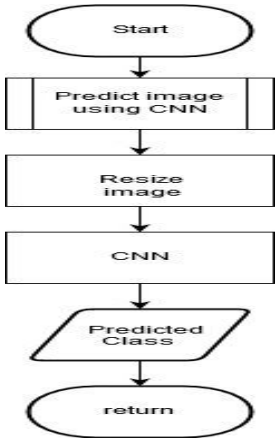
terdeteksi dengan *confidence* tertinggi. Kemudian setelah ditemukan objek dengan *confidence* tertinggi maka citra akan di-*crop* agar didapatkan seluruh bagian badan dari objek yang diinginkan. Alur dari sub-proses ini dapat dilihat pada Gambar 6.



Gambar 6. *Flowchart* Sub-proses YOLO

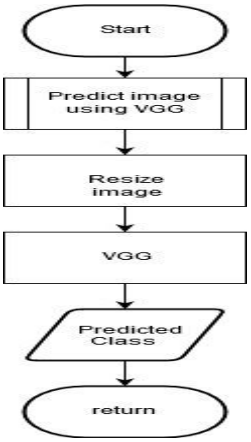
3.1.2. Convolutional Neural Network

Untuk masuk ke proses ini *image* akan terlebih dahulu diubah menjadi *grayscale*. Konversi akan dilakukan dengan mengambil nilai *value* dari nilai HSV gambar. Setelah melalui proses konversi *image* menjadi *grayscale*, *image* akan diproses dalam CNN. Pertama-tama *image* akan di-*resize* menjadi ukuran yang sesuai dan juga dipotong setengah dari ukuran keseluruhan tinggi gambar. Pada sistem ini digunakan ukuran *image* 128x128 pixel. Setelah *resize* maka *image* akan diolah menggunakan CNN lalu akan mengeluarkan hasil berupa kelas yang diprediksi. *Flowchart* dari sub-proses ini dapat dilihat pada Gambar 7.



Gambar 7. *Flowchart* Sub-proses CNN

3.1.3. VGG16



Pada proses ini, *image* yang digunakan adalah hasil potongan dari YOLO, *image* yang digunakan berbeda daripada yang digunakan untuk CNN karena model VGG tidak menerima *input image* yang berupa *grayscale*. Jadi gambar yang masuk ke model VGG berupa gambar RGB. Ukuran yang digunakan pada proses *resize* adalah 224x224 pixel. Setelah *resize* maka *image* juga akan diolah menggunakan VGG lalu akan mengeluarkan hasil berupa kelas yang diprediksi. *Flowchart* dari sub-proses ini dapat dilihat pada Gambar 8.

Gambar 8. *Flowchart* Sub-proses VGG

3.2 Desain Sistem

Pada bagian ini akan dijelaskan tentang *user interface* dari program. Di halaman awal, terdapat tempat agar *user* dapat men-*upload input*-an gambar yang ingin diketahui jenis golongannya. Halaman yang sama juga akan digunakan untuk menampilkan hasil pengenalan jenis golongan kendaraan dari *input* gambar yang di-*upload* oleh user.

User dapat memilih input gambar kendaraan yang akan dideteksi jenis golongan kendaraannya. Untuk format input gambar, hanya *file* dengan ekstensi .JPG dan .JPEG saja yang dapat dideteksi. Setelah *user* memilih gambar yang akan dideteksi, maka akan terlihat proses pengolahan dari gambar yang di-*upload* oleh *user* dari awal sampai pada hasil akhir yaitu adalah pengenalan jenis golongan kendaraan.

4. IMPLEMENTASI SISTEM

Implementasi sistem dilakukan pada computer dengan spesifikasi:

- RAM : 8GB
- STORAGE : 128 GB SSD + 1TB HDD
- CPU : Intel Core i7 6700HQ
- GPU : NVDIA GeForce GTX 960M
- OS : Windows 10 Pro

Implementasi pengkodean sistem, menggunakan Bahasa pemrograman Python dengan versi 3.6.9. Adapun beberapa *library* yang mendukung sistem ini adalah:

- OpenCV
- Numpy
- Tensorflow
- Keras
- Flask
- Scikit-learn
- Matplotlib

5. ANALISA DAN PENGUJIAN

5.1. Pengujian Sistem

Pengujian sistem digunakan untuk menilai performa dari sistem *training* yang telah dibuat. Penilaian performa dibedakan setiap jenis golongan kendaraan. Pengujian ini dilakukan pada gambar kendaraan yang telah melalui proses deteksi posisi kendaraan, konversi menjadi *grayscale* serta pemotongan dan juga *resize*.

5.1.1. Pengujian Metode CNN

Pengujian performa pada CNN akan mengevaluasi akurasi, *precision, recall* dan *f-score* pada tiap golongan. Konfigurasi yang digunakan untuk CNN adalah konfigurasi yang telah diuji yang menghasilkan akurasi terbaik. Konfigurasi yang digunakan ada dua yaitu pengukuran pertama dengan 4 *layer* serta pengukuran kedua dengan 1 *layer*. *Epoch* yang digunakan untuk uji performa adalah *epoch* 60 dan 80. Total rata-rata akurasi semua golongan kendaraan dengan konfigurasi pengukuran pertama adalah 93.51% dan rata-rata *f-score* adalah 81.37%. Sedangkan untuk konfigurasi pengukuran kedua, total rata-rata akurasi semua golongan kendaraan adalah 92.51%, 1% lebih rendah dari pengukuran pertama. Rata-rata dari *f-score* adalah 78.08%, 3% lebih rendah dari pengukuran pertama. Hal ini menunjukkan konfigurasi pengukuran pertama menghasilkan performa yang lebih tinggi dibandingkan konfigurasi pengukuran kedua. Tabel 2 menunjukkan performa model CNN dengan 4 *layer* untuk golongan satu. Tabel 3 menunjukkan performa model CNN dengan 4 *layer* untuk golongan dua. Tabel 4 menunjukkan performa model CNN dengan 4 *layer* untuk golongan tiga. Tabel 5 menunjukkan performa model CNN dengan 4 *layer* untuk golongan empat. Tabel 6 menunjukkan performa model CNN dengan 4 *layer* untuk golongan lima pengukuran. Tabel 7 menunjukkan performa model CNN dengan 1 *layer* untuk golongan satu. Tabel 8 menunjukkan performa model CNN dengan 1 *layer* untuk golongan dua. Tabel 9 menunjukkan performa model CNN dengan 1 *layer* untuk golongan tiga. Tabel 10 menunjukkan performa model CNN dengan 1 *layer* untuk golongan empat. Tabel 11 menunjukkan performa model CNN dengan 1 *layer* untuk golongan lima.

Tabel 2. Uji Performa Golongan I Metode CNN 4 *layer*

Epoch	Akurasi	Precision	Recall	F-Score
60	96.78%	95.52%	91.42%	93.43%
80	96.07%	96.82%	87.14%	91.72%

Tabel 3. Uji Performa Golongan II Metode CNN 4 *layer*

Epoch	Akurasi	Precision	Recall	F-Score
60	95%	92.42%	87.14%	89.7%
80	93.57%	82.5%	94.28%	88%

Tabel 4. Uji Performa Golongan III Metode CNN 4 *layer*

Epoch	Akurasi	Precision	Recall	F-Score
60	95.71%	83.92%	94%	88.67%
80	96.42%	91.66%	88%	89.79%

Tabel 5. Uji Performa Golongan IV Metode CNN 4 *layer*

Epoch	Akurasi	Precision	Recall	F-Score
60	91.78%	78.46%	85%	81.6%
80	95%	84.84%	93.3%	88.88%

Tabel 6. Uji Performa Golongan V Metode CNN 4 *layer*

Epoch	Akurasi	Precision	Recall	F-Score
60	92.85%	69.23%	60%	64.28%
80	94.64%	82.60%	63.33%	71.69%

Tabel 7. Uji Performa Golongan I Metode CNN 1 *layer*

Epoch	Akurasi	Precision	Recall	F-Score
60	94.64%	92.3%	85.71%	88.88%
80	95%	92.42%	87.14%	89.7%

Tabel 8. Uji Performa Golongan II Metode CNN 1 *layer*

Epoch	Akurasi	Precision	Recall	F-Score
60	90.35%	77.92%	85.71%	81.63%
80	91.78%	81.33%	87.14%	84.13%

Tabel 9. Uji Performa Golongan III Metode CNN 1 *layer*

60	93.92%	83.67%	82%	82.82%
80	94.64%	85.71%	84%	84.84%

Tabel 10. Uji Performa Golongan IV Metode CNN 1 *layer*

60	89.64%	71.83%	85%	77.86%
80	90.35%	73.23%	86.66%	79.38%

Tabel 11. Uji Performa Golongan V Metode CNN 1 *layer*

60	90.71%	61.11%	36.66%	45.83%
80	91.78%	68.42%	43.33%	53.06%

5.1.2. Pengujian Metode VGG

Pengujian performa pada VGG akan mengevaluasi akurasi, *precision, recall* dan *f-score* pada tiap golongan. Konfigurasi yang digunakan adalah konfigurasi terbaik yang telah diuji. *Epoch* yang digunakan untuk uji performa adalah *epoch* 60 dan 80. Total rata-rata akurasi semua golongan kendaraan untuk VGG adalah 90.76%, sedangkan rata-rata dari *f-score* adalah 73.53% Tabel 12 menunjukkan performa model untuk golongan satu. Tabel 13 menunjukkan performa model untuk golongan dua. Tabel 14 menunjukkan performa model untuk golongan tiga. Tabel 15 menunjukkan performa model untuk golongan empat. Tabel 16 menunjukkan performa model untuk golongan lima.

Tabel 12. Uji Performa Golongan I Model VGG

Epoch	Akurasi	Precision	Recall	F-Score
60	96.07%	90.41%	94.28%	92.3%
80	96.07%	96.82%	87.14%	91.73%

Tabel 13. Uji Performa Golongan II Model VGG

Epoch	Akurasi	Precision	Recall	F-Score
60	93.21%	91.8%	80%	85.49%
80	89.64%	72.04%	95.71%	82.2%

Tabel 14. Uji Performa Golongan III Model VGG

Epoch	Akurasi	Precision	Recall	F-Score
60	92.14%	75%	84%	79.24%
80	91.42%	90.62%	58%	70.73%

Tabel 15. Uji Performa Golongan IV Model VGG

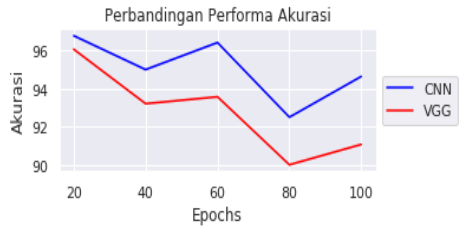
Epoch	Akurasi	Precision	Recall	F-Score
60	88.21%	65.88%	93.33%	77.24%
80	90%	75.8%	78.33%	77.04%

Tabel 16. Uji Performa Golongan V Model VGG

Epoch	Akurasi	Precision	Recall	F-Score
60	91.07%	100%	16.66%	28.57%
80	90.71%	56.66%	56.66%	56.66%

5.1.3. Perbandingan Uji Coba CNN dan VGG16

Pengujian kepada kedua model ini akan dibandingkan performanya yaitu untuk akurasi, *precision*, *recall* dan juga *f-score*. Performa yang diambil adalah performa terbaik pada *epoch* yang berbeda-beda dari setiap golongan. Untuk CNN yang diambil adalah konfigurasi yang menghasilkan performa terbaik



yaitu pada pengukuran pertama dengan 4 *layer*.

Gambar 9. Perbandingan Performa *F-Score* CNN dan VGG

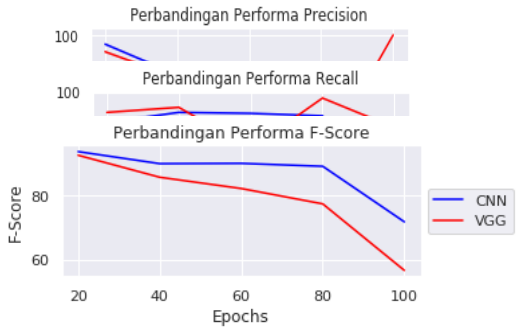
Gambar 10. Perbandingan Performa *Precision* CNN dan VGG

Gambar 11. Perbandingan Performa *Recall* CNN dan VGG

Gambar 12. Perbandingan Performa *F-Score* CNN dan VGG

5.2. Pengujian Aplikasi

Pengujian dilakukan pada *user interface* yang telah dibuat untuk



bisa *upload* gambar kendaraan dengan lebih mudah. Pengujian dilakukan dalam 2 tahap yaitu dengan menggunakan YOLO untuk mendeteksi lokasi kendaraan pada gambar dan tahap lainnya tidak menggunakan YOLO untuk mendeteksi lokasi kendaraan, tetapi gambar langsung diprediksi golongannya. Pada *user interface* ini ditampilkan semua proses yang dilalui oleh gambar sampai klasifikasi golongan jenis kendaraan.

5.2.1. *Pengujian dengan YOLO*

Pada proses ini gambar pertama-tama akan melalui proses pendeteksian posisi kendaraan pada gambar terlalu dahulu lalu baru diprediksi menggunakan *neural network* untuk mengetahui golongan kendaraan tersebut. Contoh pengujian pada golongan I dan berhasil dapat dilihat pada Gambar 13, sedangkan yang gagal

dapat dilihat pada Gambar 14.

Gambar 13. Pengujian Aplikasi Menggunakan YOLO

Gambar 14. Pengujian Aplikasi Menggunakan YOLO dan Gagal

5.2.2. *Pengujian Tanpa Menggunakan YOLO*

Pada proses ini gambar tidak akan melalui proses pendeteksian posisi kendaraan terlebih dahulu pada gambar tapi akan langsung diklasifikasikan menggunakan *neural network* untuk mengetahui golongan kendaraan tersebut. Contoh pengujian pada golongan I

dapat dilihat pada Gambar 15.

Gambar 15. Pengujian Aplikasi Tanpa YOLO

6. KESIMPULAN

Setelah dilakukan perancangan sistem, pengimplementasian, dan pengujian terhadap aplikasi yang telah dibuat, dapat ditarik kesimpulan sebagai berikut:

- Performa dari model *Convolutional Neural Network* yang konfigurasinya yang tepat dapat menghasilkan keseluruhan performa yang lebih baik yaitu akurasi sebesar 93.5% dan *f-score* sebesar 81.37% dibandingkan dengan model VGG16

yang sudah dilatih terlebih dahulu pada 1000 gambar yaitu akurasi sebesar 90.76% dan *f-score* sebesar 73.53%.

- Berdasarkan pengujian bab 5, kedua konfigurasi CNN yang diuji terlebih dahulu menghasilkan performa yang lebih baik dibandingkan VGG16.
- Deteksi posisi kendaraan(*with* YOLO) pada gambar sangat berpengaruh pada klasifikasi golongan jenis kendaraan, tanpa pencarian posisi kendaraan(*without* YOLO) maka fitur yang didapatkan menjadi sangat banyak sehingga menyulitkan *neural network* untuk mengenali jenis kendaraan.

Saran untuk pengembangan kedepannnya adalah:

- Menambah jumlah *dataset* yang lebih bervariasi agar dapat meningkatkan akurasi.
- Menguji konfigurasi lain pada CNN agar dapat meningkatkan akurasi.
- Menggunakan *pre-trained* model lainnya selain VGG16 untuk dapat mencari akurasi yang lebih tinggi.

7. DAFTAR PUSTAKA

[1] Ciaburro, G., & Venkateswaran, B. 2017. Neural Networks with R.Birmingham: Packt Publishing Ltd.

[2] Keputusan Menteri Pekerjaan Umum Nomor 370/KPTS/M/2007 tentang Penetapan Golongan Jenis Kendaraan Bermotor pada Ruas Jalan Tol yang sudah Beroperasi dan Besarnya Tarif Tol pada Beberapa Ruas Jalan Tol.

[3] Menegaz, M. (2018). Understanding YOLO. Retrieved July 19, 2019, from https://hackernoon.com/understanding-yolo-f5a74bbc7967

[4] Prabhu. (2018). Understanding of Convolutional Neural Network (CNN) — Deep Learning. Retrieved July 19, 2019, from https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

[5] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2016.91

[6] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*

[7] Udofia, U. (2018). Basic Overview of Convolutional Neural Network (CNN). Retrieved July 19, 2019, from https://medium.com/dataseries/basic-overview-of-convolutional-neural-network-cnn-4fcc7dbb4f17

[8] Vasudev. (2019). What is One Hot Encoding? Why And When do you have to use it? Retrieved July 19, 2019, from https://hackernoon.com/what-is-one-hot-encoding-why-and-when-do-you-have-to-use-it-e3c6186d008f

[9] VGG16 – Convolutional Network for Classification and Detection. (2018). Retrieved July 18, 2019, from https://neurohive.io/en/popular-networks/vgg16/

[10] Wisdom, B. D. (2019). Understanding the Confusion Matrix. Retrieved July 19, 2019, from https://dev.to/overrideveloper/understanding-the-confusion-matrix-2dk8

[11] Yu, W., Cn, W. Y. E., Yang, K., Bai, Y., Xiao, T., Yao, H., … Rui, Y. (2016). Visualizing and Comparing AlexNet and VGG using Deconvolutional Layers. *Icml*