

Deteksi Pelanggaran Parkir Pada Bahu Jalan Tol Dengan Intelligent Transportation System Menggunakan Algoritma Faster R-Cnn

Parking Violation Detection On The Roadside Of Toll Roads With Intelligent Transportation System Using Faster R-Cnn Algorithm

1st Ramadhan Admiral Hamzah
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
ramadhanhamzah@student.telkom
university.ac.id

2nd Casi Setianingsih
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
setiacasie@telkomuniversity.ac.id

3rd Ratna Astuti Nugrahaeni
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
ratnaan@telkomuniversity.ac.id

Abstrak

Pada jalan tol terdapat beberapa pengendara yang sering menggunakan bahu jalan tol sebagai lajur untuk mengemudi yang seharusnya tidak boleh digunakan kecuali keadaan mendesak, hal itu dapat mengganggu pengendara lain yang memiliki kepentingan lebih mendesak yang seharusnya menggunakan lajur tersebut, selama ini jika terjadi pelanggaran pada bahu jalan tol, petugas tol jarang mengetahui pelanggaran secara langsung, hal tersebut masih kurang efektif dikarenakan petugas tidak dapat terjun secara langsung untuk mengamankan pengemudi yang melakukan pelanggaran. Dengan menggunakan sistem yang digunakan untuk mendeteksi pelanggaran pada bahu jalan tol dengan menggunakan algoritma *Faster Region based Convolutional Neural Network (Faster R-CNN)*. Cara kerjanya dengan memproses video yang direkam dari CCTV pada aplikasi HK TOLL APPS, kemudian sistem mendeteksi kendaraan yang melakukan pelanggaran, jika terbukti melakukan pelanggaran sistem akan memberikan sebuah notifikasi kepada petugas jalan tol melalui sebuah *Bot Telegram*. Luaran yang didapatkan pada penelitian Tugas Akhir ini adalah sistem

dapat mendeteksi kendaraan yang melakukan pelanggaran pada bahu jalan tol dengan baik. Dengan menggunakan model hasil *testing* yang dilakukan dengan beberapa skema pengujian *training* partisi data dan pengujian variabel konfigurasi didapatkan model yang memiliki nilai terbaik dengan nilai *accuracy* 77.9 %, *precision* 71.1 %, *recall* 73.6 %, dan *Mean Average Precision (mAP)* sebesar 77 %.

Kata Kunci : Jalan Tol, Bahu Jalan Tol, *Faster Region based Convolutional Neural Network (Faster R-CNN)*, Deteksi Pelanggaran.

Abstract

On the roadside of the toll road there are some drivers who often use the roadside of the toll road as a lane for driving that should not be used unless in emergency, it can interfere with other drivers who have more urgent situation who should use the lane. if there is a violation on the roadside of the toll road, toll officers rarely knowing the violation directly. Using a system to detect violations on the roadside of the toll road using the Faster Region-based Convolutional Neural Network (Faster R-CNN) algorithm. How it works by processing video recorded from CCTV on the "HK TOLL APPS", the system detects vehicles that

commit violations, if proven to be a violation, the system will give a notification to toll road officers through a Telegram Bot. Output obtained in this research is the system can detect vehicles that commit violations on the roadside of the toll road. Using the model from testing results conducted with several data partition training schemes and configuration variable testing, the best model obtained with an accuracy value of 77.9 %, precision 71.1 %, recall 73.6 %, and Mean Average Precision (mAP) of 77%.

Keywords: Toll Road, Roadside of The Toll Road, Faster Region based Convolutional Neural Network (Faster R-CNN) Violation Detection.



I. PENDAHULUAN

Jalan tol merupakan jalanan umum yang digunakan oleh kendaraan, umumnya seperti mobil, bus, truk, kecuali kendaraan beroda dua seperti motor. Tujuannya adalah untuk mempersingkat jarak perjalanan pengemudi kendaraan. Menurut Peraturan Pemerintah Nomor 15 Tahun 2005 Pasal 41 [1]. Salah satu dari empat poin menjelaskan bahwa pada jalan tol terdapat bahu jalan yang penggunaannya untuk mempermudah pengemudi yang mengalami keadaan darurat saja, pengemudi yang tidak mengalami keadaan darurat tidak diperbolehkan untuk menggunakan bahu jalan. Umumnya pengemudi yang melanggar aturan jarang diketahui oleh petugas tol, dan pelanggaran peraturan hanya dapat ditindak saat petugas tol berpatroli saja.

Namun pada kenyataannya masih banyak sekali pengemudi yang melanggar aturan yang ada pada jalan tol, seperti adanya pengemudi yang berhenti dalam waktu yang lama pada bahu jalan tol. Penanganan pelanggaran aturan masih kurang efektif dikarenakan pengemudi lain harus menghubungi *call center* petugas tol agar petugas dapat mengetahui lokasi pelanggaran berada, ataupun pelanggaran peraturan hanya dapat ditindak saat petugas tol berpatroli saja.

Oleh karena itu pada penelitian ini membahas pembuatan sistem dengan menggunakan algoritma *Deep Learning Faster R-CNN*, Sistem ini bertujuan untuk mempermudah petugas jalan tol untuk mengetahui pelanggaran yang terjadi pada bahu jalan tol pada ruas tertentu dengan video CCTV milik pengelola jalan, nantinya jika terbukti terdapat kendaraan yang melanggar pada bahu jalan tol, maka sistem akan memberikan notifikasi kepada petugas jalan tol melalui sebuah *Bot Telegram* bahwa telah terjadi pelanggaran pada bahu jalan tol.

II. KAJIAN TEORI

A. Bahu Jalan Tol

Bahu jalan tol merupakan salah satu bagian penting yang ada pada jalan tol, fungsinya adalah untuk mobilisasi kendaraan saat kondisi darurat, di Indonesia peraturan mengenai kegunaan dari bahu jalan tol diatur menurut Peraturan Pemerintah Nomor 15 Tahun 2005 Pasal 41 yang terdiri dari 4 poin, yaitu: bahu

jalan digunakan bagi arus lalu lintas pada keadaan darurat, diperuntukkan bagi kendaraan yang berhenti darurat, tidak digunakan untuk menarik/menderek/ mendorong kendaraan, dan tidak digunakan untuk keperluan menaikkan atau menurunkan penumpang dan/atau barang dan/atau hewan [1].

B. *Intelligent Transportation Systems* *Intelligent Transport Systems* (ITS)

merupakan sebuah sistem jalan dan lalu lintas yang menggabungkan informasi, komunikasi, dan teknologi kontrol untuk mengintegrasikan pengemudi, kendaraan, dan jalan dengan sedemikian rupa sehingga membantu para pengemudi. Tujuannya tidak hanya untuk memecahkan masalah lalu lintas kendaraan, tetapi juga masalah sosial ekonomi, vitalitas pariwisata, dan pemanfaatan pembangunan ekonomi berkelanjutan Teknologi informasi [2].

C. Pengolahan Citra Digital

Pengolahan Citra Digital merupakan sebuah ilmu yang mempelajari cara untuk mengolah sebuah citra analog dengan tujuan untuk dapat direpresentasikan didalam komputer, dikarenakan citra analog tidak dapat diproses secara langsung ke dalam komputer maka perlu dilakukan beberapa metode sehingga citra dapat direpresentasikan kedalam komputer sehingga menjadi sebuah citra digital [3].

Citra Digital adalah sebuah array dua dimensi yang berisikan nilai real dan kompleks yang ditampilkan dengan sebuah deret bit tertentu. Suatu Citra dapat dijelaskan dengan fungsi $f(x, y)$, dengan M baris dan N kolom, dimana x, y koordinat spasialnya, serta amplitudo f pada titik koordinat (x, y) disebut dengan intensitas. Jadi sebuah Citra dapat disebut Citra Digital jika x, y serta nilai amplitudo f keseluruhannya bernilai diskrit serta berhingga [4].

Sebuah Citra Digital dapat direpresentasikan dengan bentuk Matriks sebagai berikut:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N - 1) \\ f(1,0) & f(1,1) & \dots & f(1, N - 1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M - 1,0) & f(M - 1,1) & \dots & f(M - 1, N - 1) \end{bmatrix} \quad (2.1)$$

Gambar 1. Persamaan Matriks Citra Digital [4].

D. Deep Learning

Deep Learning merupakan turunan dari *Machine Learning* yang dapat memproses dan mengolah data dengan memanfaatkan layer yang sangat kompleks. Saat ini *Deep Learning* telah secara luas diterapkan dalam domain *Artificial Intelligence* tradisional, seperti diantaranya: *Semantic Parsing*, *Natural Language Processing*, *Computer Vision* dan yang lainnya [5].

E. Convolutional Neural Network (CNN)

Convolutional Neural Network merupakan salah satu jenis model *Deep Learning* untuk memproses data yang mempunyai pola *grid pattern*, contohnya seperti sebuah citra yang terdiri dari kumpulan piksel yang dapat di baca oleh komputer dalam bentuk matriks, *Convolutional Neural Network* dirancang untuk secara otomatis dan adaptif mempelajari hierarki spasial dari sebuah *features* pada sebuah citra dari *pattern* tingkat rendah hingga *pattern* tingkat tinggi. Umumnya *Convolutional Neural Network* terdiri dari beberapa blok layer, Layer ini memiliki bermacam fungsi dasar seperti diantaranya *Normalization*, *Convolution*, *Nonlinear*, *Pooling*, dan terakhir *Fully Connected Layer* [6][7].

F. Faster Region based Convolutional Neural Network (Faster R-CNN)

Faster Region based Convolutional Neural Network atau biasa disebut dengan *Faster R-CNN*, merupakan arsitektur yang menggunakan *Convolution Neural Network* (CNN) sebagai *backbone* utamanya. *Faster R-CNN*, terdiri dari dua modul. Modul pertama merupakan sebuah *Region Proposed Network*, dan modul kedua adalah detektor *Fast R-CNN* yang menggunakan *region proposed* [8].

Region Proposed Network (RPN) mengambil citra (dengan ukuran berapa pun) sebagai *input* dan *output* satu set *object*

proposal persegi panjang. *Region Proposed Network* memetakan layer terakhir *Convolutional Neural Network* ke dimensi yang lebih rendah (256-d) ke dalam *feature map*. *Faster R CNN* di model kan dengan menggunakan *Convolutional Neural Network*.

Seluruh sistem merupakan satu jaringan terpadu untuk mendeteksi sebuah objek Pada citra tertentu. Dengan menggunakan *Neural Network*, modul *Region Proposed Network* menginformasikan ke modul *Fast R-CNN* di mana letak *region proposed* berada. *Faster R-CNN* memiliki kelebihan tingkat akurasi pendeteksian objek yang sangat tajam, namun kekurangannya jika terdapat banyak sekali objek pada satu waktu, pendeteksiannya sedikit berkurang [9].

G. Telegram

Telegram merupakan salah satu social media yang dikenal dengan performa aplikasinya yang berfokus pada keamanan, kecepatannya, dan gratis. Telegram juga aplikasi *multi-platform* yang dapat dibuka dalam waktu bersamaan, Dengan Telegram, pengguna dapat mengirim pesan, foto, video, dan file jenis apa pun (doc, zip, mp3, dll), serta membuat grup seperti beberapa fitur *social media* pada umumnya, Selain keamanannya yang bagus Telegram juga didukung dengan *Application Programming Interface* (API) yang memungkinkan penggunaan *Bot* [10].

H. Confusion Matrix

Confusion Matrix merupakan sebuah parameter performansi untuk mengukur performa dari model yang sudah dilatih. *Confusion matrix* memiliki 4 parameter, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN).

Berikut merupakan contoh tabel *Confusion Matrix* [11].

Tabel 1. Tabel Confusion Matrix

Predicted Class		Actual Class	
		Positif	Negatif
	Positif	TP	FP
Negatif	FN	TN	

Dan dibawah ini merupakan penjelasan dari 4 parameter milik *Confusion Matrix*:

- *True Positive* (TP), saat sistem dapat memprediksi positif objek melanggar dan sebenarnya memang positif.
- *True Negative* (TN), saat sistem dapat memprediksi negatif objek melanggar dan sebenarnya memang negatif.
- *False Positive* (FP), saat sistem memprediksi melanggar, namun hasil sebenarnya negatif.
- *False Negative* (FN), saat sistem memprediksi tidak melanggar, namun hasil sebenarnya positif.

I. Accuracy

Accuracy merupakan nilai rasio keseluruhan bahwa prediksi benar melanggar dan tidak melanggar pada keseluruhan data. *Accuracy* menjelaskan berapa persen kelas objek yang diprediksi dengan keadaan sebenarnya.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (2.2)$$

Keterangan:

TP = *True Positive*

FP = *False Positive*

TN = *True Negative*

FN = *False Negative*

J. Precision

Precision adalah sebuah rasio prediksi benar memprediksi positif dibanding dengan seluruh hasil prediksi positif.

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (2.3)$$

Keterangan:

TP = *True Positive*

FP = *False Positive*

TN = *True Negative*

FN = *False Negative*

K. Recall

Recall adalah nilai rasio prediksi benar memprediksi positif dibanding dengan seluruh data yang sebenarnya positif.

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (2.4)$$

Keterangan:

TP = *True Positive*

FP = *False Positive*

TN = *True Negative*

FN = *False Negative*

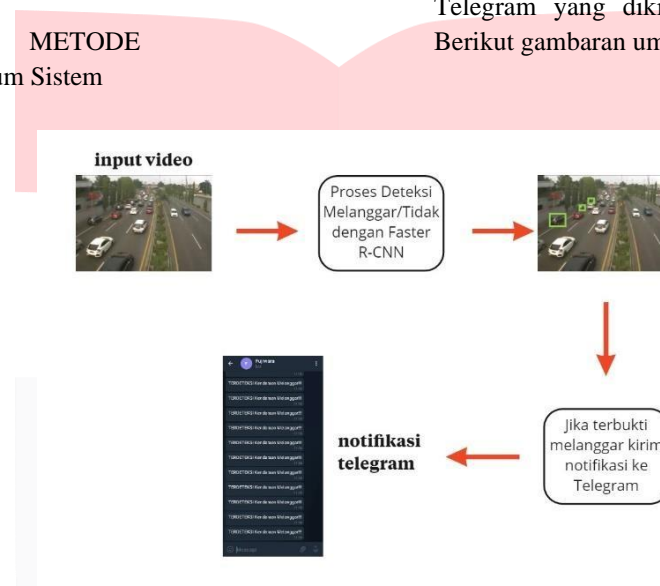
L. Mean Average Precision (mAP)

Mean Average Precision (mAP) Merupakan nilai rata-rata dari *Average Precision* (AP) adalah menemukan area yang beririsan pada kurva *precision* dan *recall*. Cara mencari mAP adalah dengan menghitung semua nilai rata-rata dari *Average Precision* [12].

Desain umum sistem ini dirancang untuk mendeteksi pelanggaran pada bahu jalan tol, sistem akan membedakan berdasarkan kelas kendaraan mana yang melanggar dan tidak melanggar, jika sistem mendeteksi kendaraan ternyata melanggar pada bahu jalan tol maka sistem akan memberikan notifikasi kepada petugas tol, melalui notifikasi yang muncul dari Telegram yang dikirim oleh *bot* Telegram. Berikut gambaran umumnya:

III. METODE

A. Gambaran Umum Sistem



Gambar 2. Gambaran Umum Sistem

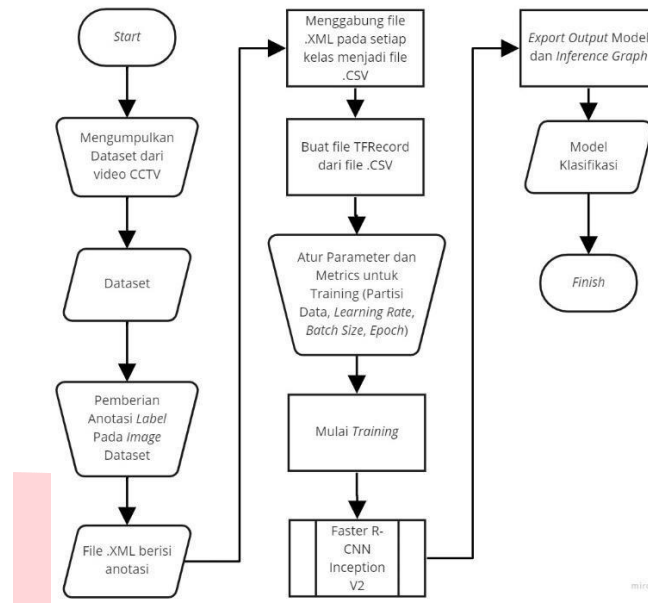
Pada desain gambaran umum sistem ini, *input* dapat berupa rekaman video keadaan CCTV ruas tol JORR-S KM 30+300, yang didapat dari aplikasi HK TOLL APPS, nantinya sistem akan mendeteksi mana kendaraan yang melanggar, jika terbukti melanggar maka *script bot* Telegram akan mengecek pembuktian kelas melanggar terdeteksi lalu akan mengirim notifikasi ke Telegram petugas tol.

B. Diagram Alur pembuatan Model

Agar sistem dapat mengetahui mana pengemudi yang melanggar ataupun yang tidak melanggar, perlu untuk membuat sebuah model

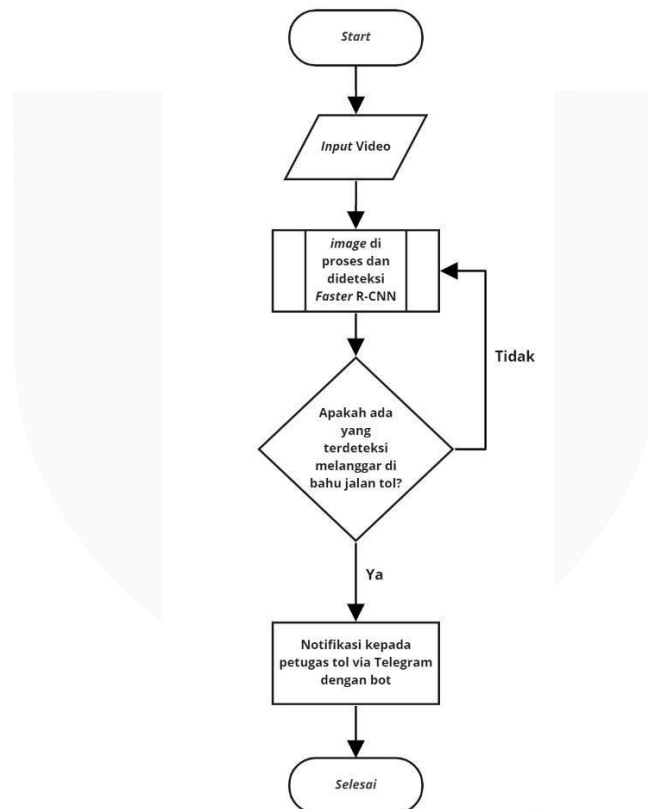
yang merupakan luaran dari proses *training* agar program dapat mengenal kelas yang ditentukan, berikut merupakan proses pembuatan model dari pencarian data hingga luaran menjadi sebuah model yang dapat dibaca oleh sistem:

- Pemberian Anotasi Label ke Dataset Gambar.
- Konversi File .XML ke .CSV.
- Konversi File .CSV ke TFRecord.
- Pembuatan Labelmap.
- Konfigurasi Variable.
- Melatih Model.
- Pembuatan *Inference Graph*



Gambar 3. Alur Pembuatan Model

C. Diagram Alur dengan Notifikasi Telegram Bot



Gambar 4. Ilustrasi Alur Kerja Sistem hingga mendapat notifikasi dari bot

Gambar diatas merupakan desain sistem pendeteksi pelanggaran pada bahu jalan tol dengan lengkap, jadi *input* berupa rekaman CCTV pada ruas jalan tol JORR-S 30+300 kemudian akan di proses oleh algoritma *Faster R-CNN* lalu setelah berhasil mendeteksi mana

kendaraan yang melanggar ataupun tidak melanggar, sistem akan mengecek jika terdeteksi kendaraan yang melanggar, maka secara otomatis akan mengirimkan notifikasi kepada *bot* Telegram lalu kemudian *bot*

Telegram akan meneruskan ke Telegram milik petugas tol.

IV. HASIL DAN ANALISIS

A. Skenario Pengujian Partisi Data

Skenario pengujian partisi data dilakukan dengan metode holdout, atau umumnya dapat disebut dengan pengujian partisi data. Untuk skenario pengujiannya, dataset akan dibagi menjadi dua jenis dan disimpan pada folder train dan test. Untuk dataset yang akan digunakan berjumlah 100 data yang terdiri dari dua kelas yaitu, melanggar yang terdiri dari 50 data dan tidak melanggar terdiri dari 50 data, nantinya akan digabung dan kemudian disatukan dengan porsi yang sudah ditentukan pada folder test dan train, lalu kemudian akan dilakukan proses melatih data atau training, yaitu sebuah proses untuk melatih data untuk dapat

mempelajari dan mendeteksi sebuah objek yang sudah ditentukan sebelumnya.

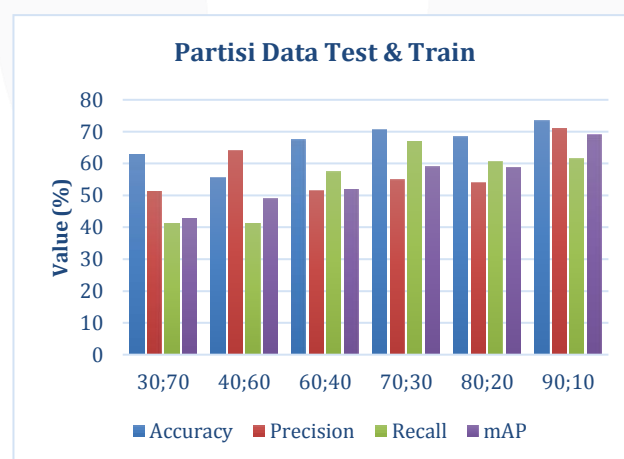
Pembagian partisi data akan diuji menjadi enam skenario training dengan rincian sebagai berikut:

1. *Train* 30% (30 data) dan *Test* 70% (70 data).
2. *Train* 40% (40 data) dan *Test* 60% (60 data).
3. *Train* 60% (60 data) dan *Test* 40% (40 data).
4. *Train* 70% (70 data) dan *Test* 30% (30 data).
5. *Train* 80% (80 data) dan *Test* 20% (20 data).
6. *Train* 90% (90 data) dan *Test* 10% (10 data).

Tujuan pada pengujian ini untuk mencari nilai *Mean Average Precision* (mAP) yang terbaik dari setiap skenario partisi data.

Tabel 2. Tabel Hasil Keseluruhan Pengujian Partisi Data

No.	Data Partition (%)	Accuracy (%)	Precision (%)	Recall (%)	Map (%)
1.	30 : 70	62.9 %	51.2 %	41 %	42.8 %
2.	40 : 60	55.5 %	64.1 %	41.1 %	49.1 %
3.	60 : 40	67.6 %	51.5 %	57.6 %	51.9 %
4.	70 : 30	70.6 %	55 %	67 %	59.2 %
5.	80 : 20	68.4 %	54 %	60.5 %	58.8 %
6.	90 : 10	73.4 %	71.1 %	61.7 %	69.1 %



Gambar 5. Grafik Total Nilai Metrics Pada Pengujian Partisi Data

Dapat diperhatikan pada grafik diatas bahwa partisi data yang terbaik setelah melakukan enam pengujian partisi data didapatkan model terbaik dengan nilai *Mean Average Precision* (mAP) paling tertinggi didapat dengan menggunakan partisi data *train*

90% dan test 10% dengan nilai *accuracy* 73.4 %, *precision* 71.1 %, *recall* 61.7 % dan mAP sebesar 69.1%, dapat disimpulkan bahwa dengan dataset dengan jumlah data yang banyak pada folder *train* dapat menghasilkan nilai akurasi dan mAP yang tertinggi dari partisi

yang lainnya sedangkan jika pada folder *test* data yang diisi dengan dataset banyak, akurasi dan mAP terlihat sangat kecil daripada sebaliknya.

B. Skenario Pengujian *Training* Variabel Konfigurasi

Pengujian *Training* variabel konfigurasi ini bertujuan untuk melihat akurasi dan performa dari hasil model yang telah selesai di *training* yang dengan merubah beberapa skema parameter pengujian variabel konfigurasi yaitu, *Learning Rate*, *Batch Size*, *Epoch*. Dengan memakai partisi data terbaik pada pengujian partisi data sebelumnya, yaitu *train* 90% dan *test* 10%. Nantinya luaran *Learning Rate* dengan *Mean Average Precision* (mAP) terbaik akan dipakai ke pengujian *Batch Size*, lalu

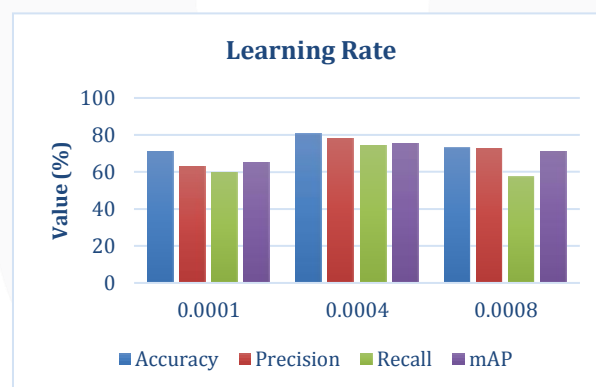
Batch Size dengan *Mean Average Precision* (mAP) terbaik akan dipakai untuk pengujian *Epoch*, Setelah itu *Mean Average Precision* (mAP) *Epoch* terbaik akan dijadikan hasil akhir pengujian dan dijadikan model hasil ujian yang paling terbaik.

C. *Learning Rate*

Pada pengujian *Learning Rate* ini parameter perbandingan untuk menghitung nilai keseluruhan dengan menggunakan *Mean Average Precision* (mAP), dengan tiga skenario *Learning Rate* yang berbeda yaitu, *Learning Rate* 0.0001, *Learning Rate* 0.0004, *Learning Rate* 0.0008. berikut merupakan hasil dari tiga skenario pengujian nilai *Learning Rate* yang berbeda.

Tabel 3. Tabel Hasil pengujian Keseluruhan *Learning Rate*

No.	<i>Learning Rate</i>	Accuracy (%)	Precision (%)	Recall (%)	Map (%)
1.	0.0001	71.3 %	62.7 %	59.6 %	65.2 %
2.	0.0004	80.6 %	77.9 %	74.1 %	75.2 %
3.	0.0008	73 %	72.8 %	57.3 %	71.1 %



Gambar 6. Grafik Total Nilai metrics dari *Learning Rate*

Dapat diperhatikan pada grafik diatas bahwa semakin tinggi nilai *Learning Rate* maka nilai *accuracy*, *precision*, *recall* dan *Mean Average Precision* (mAP) akan meningkat, namun berdasarkan 3 percobaan dengan nilai yang berbeda, *accuracy*, *precision*, *recall* dan mAP paling tertinggi didapat dengan menggunakan nilai *Learning Rate* 0.004% dengan nilai *accuracy* 80.6 %, *precision* 77.9 %, *recall* 74.1 % mAP 75.2 % dengan perbedaan *accuracy* 7.3 %, *precision* 5.1 %, *recall* 16.8 % dan mAP 4.1 %

dibanding *Learning Rate* 0.008, mengapa bukan 0.0008? dikarenakan jumlah dataset yang sedikit juga ikut mempengaruhi hasil *training* sehingga menjadi kurang stabil, seharusnya semakin tinggi nilai *Learning Rate* maka semakin tinggi juga nilai mAP yang didapatkan. Semua percobaan *Learning Rate* dilakukan dengan partisi data 90:10, dan *step* 40.000.

D. *Batch Size*

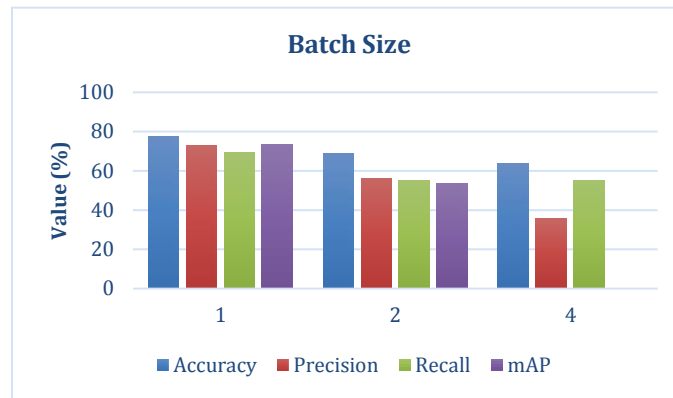
Pada pengujian *Batch Size* ini parameter perbandingan untuk menghitung

nilai keseluruhan dengan menggunakan *Mean Average Precision* (mAP), dengan tiga skenario *Batch Size* yang berbeda yaitu, *Batch Size 1*, *Batch Size 2*, *Batch Size*

4. Berikut merupakan hasil dari tiga skenario pengujian nilai *Batch Size* yang berbeda.

Tabel 4. Tabel Hasil pengujian Keseluruhan *Batch Size*

No.	<i>Batch Size</i>	<i>Accuracy</i> (%)	<i>Precision</i> (%)	<i>Recall</i> (%)	mAP (%)
1.	1	77.5 %	72.8 %	69.3 %	73.6 %
2.	2	68.8 %	55.9 %	55 %	53.8 %
3.	4	63.8 %	35.5 %	55.2 %	0.03 %



Gambar 7. Grafik Total Nilai metrics dari *Batch Size*

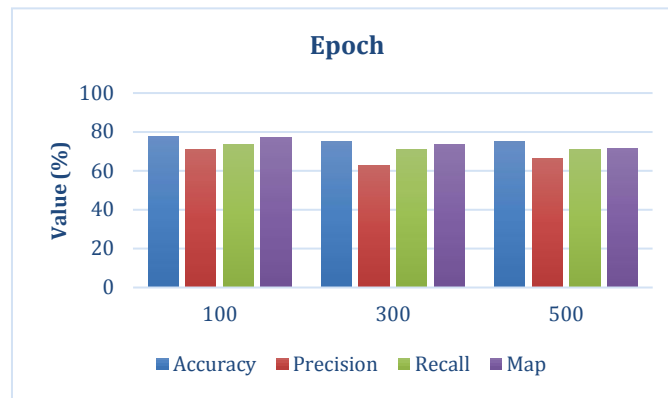
Dapat diperhatikan pada grafik diatas bahwa semakin rendah nilai *Batch Size* maka nilai *Mean Average Precision* (mAP) akan meningkat, nilai *accuracy*, *precision*, *recall* dan mAP paling tertinggi didapat dengan menggunakan nilai *Batch Size 1* dengan nilai *accuracy* 77.5 %, *precision* 72.8 %, *recall* 69.3 % dan mAP sebesar 73.6 %. Semua percobaan *Batch Size* dilakukan dengan partisi data 90:10, *Learning Rate* 0.0004, dan *step* 40.000.

E. Epoch

Pada pengujian *Epoch* ini parameter perbandingan untuk menghitung nilai keseluruhan dengan menggunakan *Mean Average Precision* (mAP), dengan tiga skenario *Epoch* yang berbeda yaitu, *Epoch 100*, *Epoch 300*, *Epoch 500*. Berikut merupakan hasil dari tiga skenario pengujian nilai *Epoch* yang berbeda.

Tabel 5. Tabel Hasil pengujian Keseluruhan *Epoch*

No.	Epoch	<i>Accuracy</i> (%)	<i>Precision</i> (%)	<i>Recall</i> (%)	mAP (%)
1.	100	77.9 %	71.1 %	73.6 %	77 %
2.	300	75 %	62.7 %	71.1 %	73.8 %
3.	500	75.3 %	66.1 %	70.9 %	71.6 %

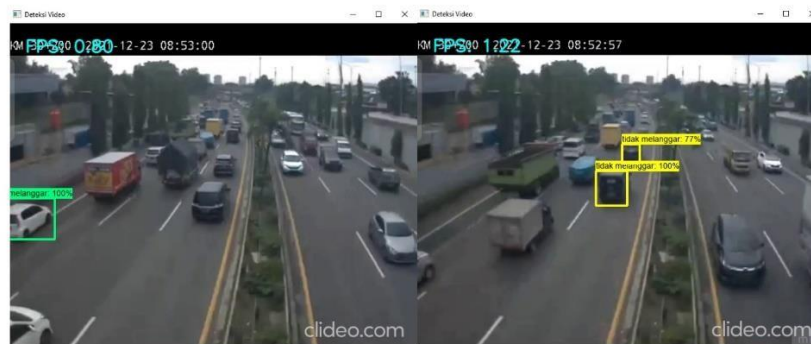


Gambar 8. Grafik Total Nilai metrics dari *Epoch*

Dapat diperhatikan pada grafik diatas bahwa semakin rendah nilai *Epoch* maka nilai *Mean Average Precision* (mAP) akan meningkat, nilai *accuracy*, *precision*, *recall* dan mAP paling tertinggi didapat dengan menggunakan nilai *Epoch* 100 dengan nilai *accuracy* 77.9 %, *precision* 71.1 %, *recall* 73.6 % mAP 77 %.

Percobaan *Epoch* dilakukan dengan partisi data 90:10, *Learning Rate* 0.0004, *Batch Size* 1 dan *step* berbeda-beda dikarenakan jumlah *step* dan *Batch Size* mempengaruhi hasil *Epoch*. *Epoch* 100 menggunakan *step* 10.000, *Epoch* 300 menggunakan *step* 30.000, *Epoch* 500 menggunakan *step* 50.000.

F. Luaran Hasil Deteksi



Gambar 9. Hasi Deteksi Melanggar (Bag. Kiri), dan Tidak Melanggar (Bag. Kanan).

Gambar diatas merupakan luaran yang didapat dari melakukan proses-proses pada sub bab sebelumnya hingga menghasilkan sebuah model. Model dapat digunakan dan dapat dicoba untuk mendeteksi kendaraan yang melanggar, berikut merupakan screenshot hasil deteksi dari sistem yang menggunakan *input* video CCTV sebagai input untuk mendeteksi gambar yang ada pada video.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

1. Dengan menggunakan sebuah sistem deteksi objek yang menggunakan *Faster R-CNN* sebagai algoritma dapat mengidentifikasi terjadinya pelanggaran pada bahu jalan tol,

setelah melalui skema pengujian partisi data dan pengujian konfigurasi variabel maka didapatkan hasil yang terbaik dengan nilai *accuracy* 77.9 %, *precision* 71.1 %, *recall* 73.6 %, dan *Mean Average Precision* (mAP) sebesar 77 %. Dengan menggunakan partisi data 90:10 dan variabel konfigurasi *Learning Rate* 0.0004, *Batch Size* 1, dan *Epoch* 100.

2. Sistem dapat mendeteksi dan memberikan notifikasi ke petugas tol dengan menggunakan *bot* Telegram yang berfungsi hanya saat sistem mendeteksi kelas melanggar saja dan semuanya dapat berjalan dengan baik.

B. Saran

1. Menambah variasi dataset saat kepadatan kendaraan ramai, sedang, lengang agar hasil yang didapat bisa lebih maksimal lagi.
2. Pada notifikasi menambahkan fitur lain seperti screenshot saat terjadi pelanggaran, serta lokasi dimana terjadinya pelanggaran.

REFERENSI

- [1]. Peraturan Pemerintah (PP) tentang Jalan Tol, Pemerintah Pusat, March, 2005. Accessed on: January. 21, 2022.[Online]. Available:<https://peraturan.bpk.go.id/Home/Details/49351/pp-no-15-tahun-2005>.
- [2]. Intelligent Transport Systems (ITS) Introduction Guide, version 1.1, Japan Society of Civil Engineers (JSCE), August 2016. Accessed on: January. 21, 2022 [Online]. Available: https://www.jsceint.org/system/files/ITS_Introduction_Guide_2.pdf.
- [3]. PN Andono, T Sutojo, Pengolahan Citra Digital, 1st ed. Andi Publisher, 2017.
- [4]. D. Putra, Pengolahan Citra Digital, 1st ed. Andi Publisher, 2010.
- [5]. Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, Michael S. Lew, "Deep learning for visual understanding: A review", Elsevier B.V., pp. 1, 2015.
- [6]. Yamashita, R., Nishio, M., Do, R.K.G. et al. Convolutional neural networks: an overview and application in radiology. *Insights Imaging* 9, 611–629 (2018). <https://doi.org/10.1007/s13244-018-0639-9>.
- [7]. Salman Khan; Hossein Rahmani; Syed Afaq Ali Shah; Mohammed Bennamoun; Gerard Medioni; Sven Dickinson, *A Guide to Convolutional Neural Networks for Computer Vision*, Morgan & Claypool, 2018.
- [8]. Shaoqing Ren, Kaiming He, Ross Girshick, "Faster R-CNN: Towards RealTime Object Detection with Region Proposal Networks", *IEEE Transactions on Pattern Analysis and Machine Intelligence* Volume: 39, Issue: 6, June 1, 2017, pp. 3, 2017.
- [9]. Ming-Jui Ho, Yu-Chin Lin, Hsin-Chuan Hsu, "An Efficient Recognition Method for Watermelon Using Faster R-CNN with Post-Processing", 2019 8th International Conference on Innovation, Communication and Engineering (ICICE), pp. 86, 2019.
- [10]. Telegram, "Telegram FAQ", telegram.org. Accessed on: January. 31, 2022. Available: <https://telegram.org/faq#q-what-is-telegram-what-do-i-do-here>.
- [11]. Kevin Adiputra Shianto, Kartika Gunadi, Endang Setyati, "Deteksi Jenis Mobil Menggunakan Metode YOLO Dan Faster R-CNN", *Jurnal Informatik: Jurnal Pengembangan IT (JPIT)*, Vol.07, No.01, 2019, pp. 2-3, 2019.
- [12]. Shivy Yohanandan, "mAP (mean Average Precision) might confuse you!", towardsdatascience.com. Accessed on: February. 04, 2022. Available: <https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2>.