

Deteksi Pelanggaran Parkir Pada Bahu Jalan Tol Dengan Intelligent Transportation System Menggunakan Algoritma Yolo

(Parking Violation Detection On The Roadside Of Toll Roads With Intelligent Transportation System Using Yolo Algorithm)

1st Maulana Heardy Yusfian
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

maulanahy@student.telkomuniversi
ty.ac.id

2nd Casi Setianingsih
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

setiacasie@telkomuniversity.ac.id

3rd Ratna Astuti
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

ratnaan@telkomuniversity.ac.id

Abstrak

Jalan Tol merupakan Jalan bebas hambatan dan salah satu sarana yang banyak digunakan oleh masyarakat umum untuk mempersingkat waktu dalam bepergian. Sebagai jalan bebas hambatan, jalan tol mempunyai bahu jalan yang dapat digunakan untuk kepentingan khusus seperti Ambulance, Pemadam Kebakaran, atau keadaan darurat lainnya. Seringkali bahu jalan disalahgunakan untuk kepentingan yang tidak mendesak dan tidak darurat sehingga dapat merugikan pengguna jalan tol lainnya. Hal ini disebabkan kurangnya pengawasan dari petugas jalan Tol.mengatasi permasalahan tersebut. Sistem ini memiliki komponen utama, yaitu: algoritma YOLOV4 deteksi objek kendaraan. Kemudian kamera yang disebar di area sekitar bahu jalan tol agar dapat merekam pergerakan kendaraan. Hasil rekam data tersebut dapat diolah menjadi informasi yang akan di kirim ke petugas tol terdekat melalui Telegram Bot. Hasil yang didapatkan melalui tugas akhir ini adalah sistem yang akan mendeteksi pelanggaran pada bahu jalan tol menggunakan model dengan rasio 90%:10%, learning rate 0.06, dan max batch64, dengan nilai mAP mencapai 97,96% yang didapatkan melalui perhitungan *confusion matrix*. Nilai akurasi yang didapatkan melalui pengujian dengan algoritma YOLOV4 (*You Only Look Once*) pada kasus ini adalah 80%.

Kata Kunci: Deep Learning, Deteksi Kendaraan, Telegram Bot, YOLOV4

Abstract

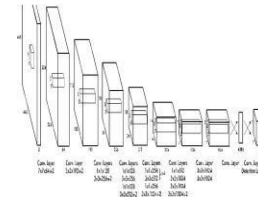
Toll road is a freeway and widely used by the general public to shorten travel time. As a freeway, a toll road has a shoulder used for special purposes such as an ambulance, fire department, or emergency situations. Often shoulder of the road is misused for non-urgent and non-emergency purposes so that it can harm other toll road users. This is due to the lack of supervision from toll road officers. This system main components, namely: the YOLOV4 algorithm for vehicle object detection. Cameras are deployed in the area around the shoulder of the toll road so that they can record vehicle movements. The results of the data recording can be processed into information that will be sent to the nearest toll road officer via Telegram Bot. The results obtained through this final project is a system that will detect violations on the toll road shoulder using a model with a ratio of 90%:10%, learning rate 0.06, and max batch64, with mAP values reaching 97.96% obtained through confusion matrix calculations. The accuracy value obtained through testing with the YOLOV4 (*You Only Look Once*) algorithm in this case is 80%.

Keywords: Deep Learning, Vehicle Detection, Telegram Bot, YOLOV4

I. PENDAHULUAN

Transportasi manusia tumbuh lebih cepat. Salah satu sarana yang mendukung pesatnya pertumbuhan transportasi adalah adanya jalan tol atau biasa disebut jalan tol. Merujuk pada peraturan nomor 15 tahun 2005 tentang jalan tol, di antaranya terdapat aturan sebagai berikut: Dilarang menarik/menarik/mendorong kendaraan lain, kecuali penarik/pendorong dari pengelola jalan tol. Dilarang menaikkan atau menurunkan penumpang dan/atau barang dan/atau hewan. Dilarang mendahului kendaraan melalui bahu jalan tol. Dilarang melintasi median dengan kendaraan untuk memotong jalur (belok kembali) [1]. *Intelligent Transportation System* dalam bahasa Indonesia berarti sistem transportasi yang cerdas. Sistem ini memiliki tujuan dasar untuk menciptakan sistem transportasi yang memiliki kecerdasan sehingga dapat membantu pengguna transportasi untuk mendapatkan informasi, mempermudah transaksi, meningkatkan kapasitas prasarana dan sarana transportasi, mengurangi kemacetan atau antrian, meningkatkan keamanan dan kenyamanan, mengurangi pencemaran lingkungan, membuatnya lebih efisien. manajemen transportasi [2]. Algoritme YOLOV4 (*You Only Look Once*) dapat mendeteksi objek waktu nyata, dan mendeteksi perangkat di kamera melalui desain kompilasi-kompresi bersama. Untuk meningkatkan efisiensi komputasi pada perangkat, skema kolaboratif GPU-CPU diadopsi bersama dengan pengoptimalan berbantuan kompilator tingkat lanjut. Di antara cakupan ITS di atas, sistem manajemen lalu lintas merupakan salah satu subsistem yang dapat dikembangkan untuk mengatasi masalah tersebut, dipadukan dengan pemanfaatan algoritma yolov4 (*You Only Look Once*) dalam teknik pendeteksian objek. Sistem ini digunakan untuk dapat membantu Petugas Jalan Tol memantau pelanggaran parkir di bahu jalan tol. Dengan menggunakan algoritma YOLOV4 untuk mengkurasi objek melalui kamera untuk memberikan informasi tentang kendaraan yang parkir sembarangan. Kamera dipasang di beberapa titik di sepanjang jalan tol yang termasuk dalam kawasan terlarang untuk merekam data kendaraan yang parkir sembarangan di kawasan tersebut. Selanjutnya data hasil perekaman dikirimkan ke aplikasi Telegram menggunakan bantuan bot telegram yang dapat diakses melalui handphone dan komputer/laptop petugas jalan tol.

II. KAJIAN TEORI

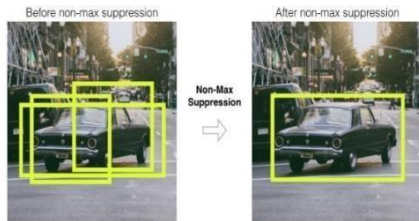


Gambar 1. *Architecture* YOLO

Berdasarkan Gambar 1 di atas, arsitektur YOLO memiliki jaringan pengenalan yang terdiri dari 24 lapisan *konvolusi* diikuti oleh 2 lapisan yang terhubung. YOLO dibuat oleh Joseph Redmon pada tahun 2015. Algoritma ini memiliki tingkat akurasi yang tinggi dengan tetap mempertahankan kecepatan proses yang tinggi serta YOLO berkembang dari jaringan YOLO dan YOLOV2. Jaringan di YOLO membagi setiap gambar dalam set pelatihan menjadi kisi $S \times S$ ($S = 14$). Jika pusat kebenaran dasar target ada di 8 grid, maka grid bertanggung jawab untuk mendeteksi target. Setiap grid memprediksi bounding box B dan skor kepercayaan, serta probabilitas bersyarat kelas C [3]. Menggunakan metode *Non-Maximum Suppression* (NMS) untuk memilih kotak pembatas terbaik. YOLO memberikan kecepatan yang lebih tinggi dibandingkan dengan *Faster-RCNN*, tetapi YOLO memiliki banyak kesalahan deteksi. Untuk mengatasi masalah ini, YOLO-V2 memperkenalkan ide "kotak jangkar" di *Faster R-CNN* dan menggunakan metode pengelompokan *k-means* untuk menghasilkan kotak pemisah sebelumnya yang sesuai. Dengan demikian, jumlah kotak jangkar yang dibutuhkan untuk mencapai hasil yang sama dengan pengurangan persimpangan atas penyatuan (IOU). YOLO-V2 meningkatkan struktur jaringan dan menggunakan lapisan konvolusi untuk menggantikan lapisan yang terhubung sepenuhnya di lapisan keluaran yolo. YOLO-V2 juga memperkenalkan normalisasi batch, klasifikasi resolusi tinggi, pengelompokan dimensi, prediksi lokasi langsung, fitur berbutir halus, pelatihan multi-skala, dan metode lain yang sangat meningkatkan akurasi deteksi dibandingkan dengan YOLO. Yolo-V3 adalah versi perbaikan dari YOLO V2. Ini menggunakan prediksi multi-skala untuk mendeteksi target akhir, dan struktur jaringannya lebih kompleks daripada YOLO-V2. YOLO V3 memprediksi kotak pembatas pada skala yang berbeda, dan prediksi multi-skala membuat YOLO-V3 lebih efektif dalam mendeteksi target kecil daripada YOLO-V2. [4]

A. Non-Max Suppression

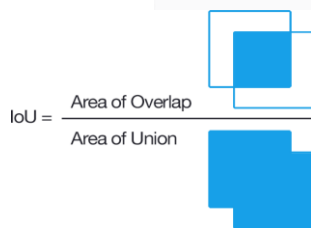
Non-Max Suppression menghilangkan kotak pembatas dari semua kotak pembatas prediktif yang ada dengan membandingkan nilai kepercayaan. Kotak pembatas yang diprediksi dengan nilai kepercayaan terendah dihapus, meninggalkan kotak pembatas dengan nilai kepercayaan tertinggi. Diagram penekanan *Non-Max Suppression* dapat dilihat pada gambar di bawah ini



Gambar 2. Non-max suppression

B. Intersection Of Union

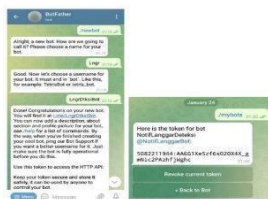
IoU adalah matriks evaluasi yang digunakan untuk mengukur keakuratan detektor objek dalam kumpulan data [6]. Jaringan IoU akan memprediksi IoU antara kotak pembatas dan kebenaran dasar yang serupa. Pada dataset pelatihan untuk model yang akan digunakan untuk tugas akhir ini, nilai *threshold* IoU yang digunakan adalah 0,5. Jika nilai IoU melebihi,



Gambar 3. IoU

a. Designing Telegram Bot

Telegram Bot berfungsi sebagai penerima pesan jika algoritma YOLOv4 pada perangkat menangkap pelanggaran lalu lintas yang dilakukan oleh warga yang menyeberang jalan. Desainnya menggunakan Bot *Father*, kerangka otomatisasi universal yang digunakan untuk membuat bot hanya dengan mengisi kuesioner yang disediakan oleh Bot *Fathers*.



Gambar 4. Cara Membuat Bot Fathers (Kiri), Token Dari Bot Telegram (Kanan)

Setelah berhasil membuat bot dengan nama yang telah didesain, maka Bot *Father* akan mengirimkan notifikasi berupa token yang digunakan untuk memanggil bot yang telah dibuat pada sistem pendeteksi pelanggaran.

III. METODE

A. Data Collection

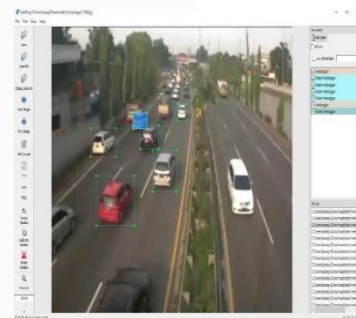
Dataset yang digunakan adalah dataset yang diambil dari aplikasi APPS TOL HK. Dalam perancangan tugas akhir ini akan diambil 100 citra secara acak, yang akan digunakan untuk pelatihan model klasifikasi. Gambar yang diambil adalah gambar kendaraan yang melintasi bahu jalan tol



Gambar 5. Dataset From HK TOLL APPS

Dataset yang digunakan adalah dataset yang diambil dari aplikasi APPS TOL HK. Dalam perancangan tugas akhir ini akan diambil 100 citra secara acak, yang akan digunakan untuk pelatihan model klasifikasi. Gambar yang diambil adalah gambar kendaraan yang melintasi bahu jalan tol.

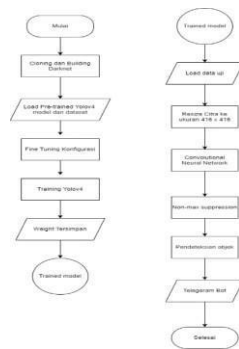
B. Image Annotation



Gambar 6. Alat LabelImg

Setelah menyiapkan dataset saatnya untuk membuat label. Anotasi adalah proses pembuatan label dengan memberikan kotak pembatas beserta nama kelas untuk objek di setiap gambar. Beri anotasi pada gambar menggunakan alat LabelImg. Untuk menentukan melanggar dan tidak dataset yang telah dikumpulkan, maka jalankan program dan buka folder tempat penyimpanan gambar. Kemudian berikan kotak batas dengan menyeret kursor dan memberi label sesuai objek. Hasil anotasi berupa data yang berisi informasi letak kotak batas dan labelnya dalam bentuk file bobot.

C. System Workflow Design



Gambar 7. Flowchart Alur Kerja Sistem Figure

Sebelum pelatihan, *CSPDarknet-53* dikloning dan dibangun sebagai arsitektur YOLOv4. Kemudian, penyesuaian hyperparameter dibuat di file konfigurasi dan setelah itu data pelatihan dapat dimulai. File bobot yang merupakan *output* dari proses pelatihan langsung disimpan di penyimpanan setelah pelatihan selesai. File bobot hasil pelatihan memiliki peran penting dalam keberhasilan sistem.

D. Hyperparameter Configuration

a. Confusion Matrix:

Pada tabel 1 terdapat tabel kolom objek yaitu kelas data yang digunakan, serta kolom hasil klasifikasi. Jika mobil yang melanggar terdeteksi di jalan tol dan sistem dapat mendeteksi pelanggaran yang benar, maka akan dikelompokkan menjadi *True Positive* karena hasil klasifikasi sesuai dengan yang diharapkan. Jika objek tersebut bukan kendaraan yang melanggar tetapi sistem mengidentifikasinya sebagai kendaraan yang tidak melanggar bahu jalan tol, maka akan dikelompokkan menjadi *False Positive*, karena hasil pendeteksian tidak sesuai dengan yang diharapkan. Pengelompokan tersebut didasarkan pada adanya objek kendaraan yang melanggar, namun sistem tidak mendeteksi adanya kendaraan yang melanggar. Dan jika objek yang bukan kendaraan melanggar terdeteksi sebagai tidak melanggar maka akan dikelompokkan menjadi *True Negative*.

Tabel 1. Confusion Matrix

Predicted Class	Actual Class	
	Positive	Negative
Positive	TP	FP
Negatif	FN	TN

b. Akurasi

Akurasi adalah parameter prediktif yang menentukan tingkat keandalan dan keberhasilan algoritma dalam mendeteksi objek pada seluruh data. Akurasi menggambarkan berapa persentase kelas objek yang benar diprediksi dan salah dari keseluruhan objek. [6]

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (1)$$

c. Presisi

Presisi adalah rasio prediksi positif sejati dibandingkan dengan hasil prediksi positif keseluruhan. Dapat dilihat pada persamaan (2.2) jika nilai *false positive* semakin tinggi maka nilai presisi semakin rendah dan sebaliknya.

$$\text{Precision} = \frac{TP}{TP+FP} \times 100\% \quad (2)$$

d. Recall (Sensitivitas)

Recall adalah rasio prediksi positif benar dibandingkan dengan semua data positif benar. Ingat dibandingkan dengan seluruh objek yang sebenarnya. *Recall* adalah rasio prediksi dengan nilai positif benar dengan semua data atau objek yang terdeteksi positif benar. Persamaan untuk mencari nilai *recall* dapat dilihat pada persamaan (2.3). Nilai *recall* yang tinggi menunjukkan bahwa sistem dapat mengklasifikasikan kelas-kelas objek dengan benar.

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\% \quad (3)$$

e. F1 Score

F1 Score adalah perbandingan rata-rata dari nilai presisi dan daya ingat.

$$F1score = 2 \times \frac{\text{Recall} \times \text{Presisi}}{\text{Recall} + \text{Presisi}} \times 100\%$$

Hal ini terlihat jika *F1 Score* semakin mendekati 1 yang menandakan bahwa kinerja sistem sudah baik. Persamaan *F1 Score* dapat dilihat dari persamaan (2.5).

f. Mean Average Precision

Nilai MAP adalah metrik yang mengevaluasi kinerja model deteksi objek. Sebelum menghitung mAP perlu dilakukan penyesuaian nilai threshold pada IoU untuk menentukan validasi objek yang terdeteksi. Nilai mAP tergantung dari nilai ambang batas IoU, untuk itu perlu adanya penyesuaian nilai ambang batas.

IV. HASIL DAN PEMBAHASAN

A. Pengujian Rasio Dataset

Setelah dilakukan pelatihan dengan membagi dataset ratio mulai dari 70% data latih : 30% data uji, 80% data latih : 20% data uji, dan 90% data latih : 10% data uji maka dapat diketahui bahwa rasio 90 % data pelatihan: 10% data pengujian menghasilkan model dengan performa terbaik. Ini karena semakin banyak data yang dilatih, semakin baik model dalam pembelajaran mesin. Hasil visual dapat dilihat pada tabel dan grafik di bawah ini. Ukuran batch = 64

Tabel 2 Pengujian Rasio

Rasio Dataset	Presisi	Recall	F1 Score	Average IoU	Average Loss	mAP	Akurasi
90% : 10%	61%	63%	62%	44.61%	1	51.93%	73%
80% : 20%	35%	59%	44%	26.47%	1	32.34%	80%
70% : 30%	39%	54%	45%	29.58%	1	26.18%	79%

Gambar 8 Hasil Pelatihan Grafik Dataset Ratio



B. Batchsize Testing

Pada pelatihan dengan Batchsize 4, 6, dan 8, hasil akurasi tertinggi adalah 77% pada Batchsize 4. Hal ini karena semakin banyak data yang dilatih, baik. Hasil visualnya dapat dilihat pada tabel di maka model dalam machine learning akan semakin bawah ini.

Tabel 3 Tabel Batchsize

Batchsize	Presisi	Recall	F1 Score	Average IoU	Average Loss	mAP	Akurasi
4	82%	95%	88%	67.52%	1	95.16%	77%
Batchsize	Presisi	Recall	F1 Score	Average IoU	Average Loss	mAP	Akurasi
6	82%	91%	86%	68.38%	1	95.64%	76%
8	85%	91%	88%	70.74%	1	96.40%	76%



Gambar 9 Hasil Grafik Batchsize

C. Learning Rate Testing

Pada pelatihan dengan *learning rate* 0,04, 0,06, dan 0,08 hasil akurasi tertinggi yaitu 80% pada *learning rate* 0,06 dan 0,08. Selisihnya adalah nilai mAP pada *learning rate* sebesar 0,06. Ini karena semakin banyak data yang dilatih, semakin baik model dalam pembelajaran mesin. Hasil visual dapat dilihat pada tabel di bawah ini.

Tabel 4 Pengujian Learning Rate

Learning Rate	Presisi	Recall	F1 Score	Average IoU	Average Loss	mAP	Akurasi
0.04	83%	93%	87%	69.26%	1	96.12%	77%
0.06	83%	96%	89%	69.45%	1	97.96%	80%
0.08	83%	96%	89%	70.05%	1	96.94%	80%



Gambar 10 Hasil Grafik Learning Rate

D. System Testing Scenario



Gambar 11 Skenario Pengujian

Video lalu lintas yang direkam oleh webcam akan segera diproses untuk mendeteksi kendaraan yang melanggar dan tidak. Deteksi menggunakan program yang telah dibuat menggunakan YOLOv4 dengan bahasa python. Pengujian yang telah dilakukan seperti pada gambar, terbukti sistem mampu mendeteksi kendaraan yang melintas dipinggir jalan tol menggunakan video yang diambil melalui aplikasi APPS TOL HK dengan menyediakan kotak pembatas. Jika positif mengidentifikasi kendaraan yang melanggar, sistem akan segera mengambil tangkapan layar otomatis dan mengirimkannya ke petugas jalan melalui bot telegram.

V. KESIMPULAN

Sistem Deteksi Pelanggaran Parkir Pada Bahu Jalan Tol Dengan *Intelligent Transportation System* Menggunakan Algoritma (Yolo) dapat mendeteksi pelanggaran pada bahu jalan tol dengan hasil *training* yang terbaik pada *learning rate* 0.06 dengan *max batches* 4000 serta nilai mAP 97.96% dan akurasi yang dihasilkan mencapai 80%. Sistem dapat mendeteksi dan memberikan pemberitahuan kepada petugas jalan tol dengan menggunakan bot telegram dengan pesan terkirim "Terdeteksi Kendaraan Melanggar". Menggunakan Jupyter Notebook pada saat training. Jika menggunakan google colab yang mempunyai limit penggunaan GPU dan terjadi limit penggunaan gpu saat melakukan training, dapat menyebabkan runtime terputus secara otomatis. Ketika sudah mencapai limit, grafik tidak akan ditampilkan sebagai hasil performansi dari training. Menggunakan perangkat keras dengan spesifikasi tinggi dengan GPU NVIDIA untuk memperlancar proses bounding box mendekteksi adanya kendaraan yang melanggar pada bahu jalan tol dan mencegah fps turun ketika[7] system mengirim pesan pada telegram bot.

REFERENSI

- [1] M. Adika Faris Ihsan. 'ini-daftar-larangan-yang-berlaku-di-jalan-tol-melanggar-bisa-didenda-rp. Diakses 05 Februari 2022 [online].
”<https://otomotif.kompas.com/read/2021/04/02/071200915/>.
- [2] G. Aisyah and P. Bestari, "Implementation E-Tilang in Bandung to Increase Awareness of Cross as Moral Law Passed Citizenship (Civic Virtue)," vol. 251, no. 22, pp. 664–667, 2018, doi: 10.2991/acec-18.2018.148.
- [3] Tian, Y., Yang, G., Wang, Z., Wang, H., Li*, E., & Liang, Z. (2019). Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Computers and Electronics in Agriculture*, 417-426.
- [4] Tian, Y., Yang, G., Wang, Z., Wang, H., Li*, E., & Liang, Z. (2019). Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Computers and Electronics in Agriculture*, 417-426.
- [5] Jain, Harshil & Nandy, S. (2019). Incremental Training for Image Classification of Unseen Objects. 10.13140/RG.2.2.10266.47046.
- [6] Kusuma, T. A., Usman, K., & Saidah, S. (2021). People counting for public transportations using you only look once method. *Jurnal Teknik Informatika (Jutif)*, 2(1), 57–66.
<https://doi.org/10.20884/1.jutif.2021.2.2.77>
<https://datahacker.rs/deep-learning-intersection-over-union/>. [Diakses 5 Januari 2021]