

Deteksi Mobil Dan Klasifikasi Jenis Mobil Menggunakan Model *Machine Learning* Untuk Sistem Parkir Pintar

1st Naufal Riz Kifli
Telkom University
Fakultas Teknik Elektro
S1 Teknik Telekomunikasi
Bandung, Indonesia

naufalrizkifli@student.telkomuniversity.ac.id

2nd Aryandhika Ibnu Raihan
Telkom University
Fakultas Teknik Elektro
S1 Teknik Telekomunikasi
Bandung, Indonesia

aryandhika@student.telkomuniversity.ac.id

Abstrak — Keterbatasan lahan parkir disertai dengan peningkatan jumlah kendaraan menyebabkan masalah-masalah kemacetan, menghabiskan waktu mencari tempat parkir, dan tempat parkir kosong yang sia-sia. Agar masalah tersebut dapat diatasi, sistem parkir pintar digunakan untuk mengatasi masalah tersebut agar pengemudi mudah dalam mencari tempat parkir sehingga tidak terjadi kemacetan dan tidak terdapat slot kosong yang sia-sia. Salah satu subsistem dalam sistem parkir pintar adalah sistem pendeteksian ketersediaan slot. Paper ini membahas bagian sistem tersebut, khususnya pada deteksi objek mobil dari gambar, agar data hasil deteksi dan klasifikasi tersebut dapat digunakan untuk deteksi ketersediaan slot dan fitur seperti reservasi. Model yang digunakan pada paper ini untuk mendeteksi mobil dan klasifikasi jenis mobil yaitu model YOLOv8s. Model tersebut dilatih dengan menggunakan data gambar yang diambil dari tempat parkir luar gedung TULT Telkom University dengan jumlah data gambar yang terbatas. *Hyperparameter* pada model YOLOv8s kami ubah yaitu *weight decay* dan *freeze*. Nilai performa terbaik didapatkan dengan nilai *weight decay* 0.1 lalu melatih lagi dengan *freeze 4 layer*, dimana nilai mAP, F1-Score, dan kecepatan pemrosesan data gambar yaitu 99.2%, 98%, dan 1.95 milidetik secara berurutan. Namun model terbaik tersebut belum tentu baik dalam mendeteksi dan mengklasifikasi objek mobil pada data baru.

Kata kunci— YOLOv8s, Deteksi, Klasifikasi, *Weight Decay*

I. PENDAHULUAN

Lahan parkir yang terbatas serta peningkatan jumlah kendaraan, menimbulkan masalah-masalah dalam aktivitas melakukan parkir kendaraan [1]. Masalah tersebut pun terjadi pada lingkungan tempat parkir Telkom University, tepatnya di luar gedung TULT. Contoh dari masalah-masalah tersebut yaitu pengendara harus menghabiskan waktunya untuk mencari tempat parkir. Kemacetan juga ditimbulkan karena pengendara harus mengantri ketika mencari tempat parkir. Selain itu, slot parkir kosong dapat tidak ditemukan oleh pengendara secara sia-sia karena tersembunyi dari pandangan pengendara.

Sistem parkir pintar yang berkembang menggunakan teknologi-teknologi IoT dapat menjadi solusi terhadap masalah dalam aktivitas parkir kendaraan. Salah satu komponen dalam parkir pintar yaitu sistem pendeteksi ketersediaan slot parkir yang berbeda-beda yang terdiri dari perangkat keras untuk mengambil data dan proses pengolahan data tersebut untuk mengetahui ketersediaan slot parkir.

Kondisi lingkungan tempat parkir mobil seperti banyaknya slot parkir mobil yang harus dicakup, jenis tempat parkir mobil (*indoor/outdoor*), dan cuaca di tempat parkir mempengaruhi pemilihan metode pendeteksian ketersediaan slot yang akan digunakan [2]. Pada paper ini metode yang digunakan adalah teknik pengolahan sinyal gambar dari kamera untuk mendeteksi mobil dan klasifikasi jenis mobil. Pengolahan sinyal gambar dari kamera di luar ruangan dapat terpengaruh oleh dampak cuaca, namun hal tersebut dapat diatasi dengan memperbaiki model pengolahan sinyal gambar [3]. Deteksi mobil menggunakan pengolahan sinyal gambar dari kamera memiliki biaya penerapan dan pemeliharaan yang mahal [4], namun lebihnya yaitu satu sensor kamera dapat menganalisis lebih dari satu slot parkir sehingga lebih efektif dalam hal pembiayaan pada area tempat parkir yang luas [5]. Selain itu pengolahan sinyal gambar dari hasil tangkapan kamera dapat menghasilkan fitur lebih banyak dibandingkan dengan metode lainnya [3].

Beberapa model *machine learning* telah digunakan dalam pengolahan sinyal gambar untuk mendeteksi objek mobil dan klasifikasi jenis mobil dari gambar [6][7][8][9]. Faster R-CNN dilatih dan divalidasi pada dataset KITTI dengan menggunakan ResNet mendapatkan nilai skor F1 sebesar 76,9% dengan mengubah parameter *learning rate* menjadi 0,0001 [7]. Selain itu model Faster R-CNN tersebut dibandingkan dengan YOLOv3 dan YOLOv5, dimana model YOLOv3 dan YOLOv5 menghasilkan nilai mAP sebesar 56% dan 84,81% secara berurutan. Waktu pemrosesan terpendek dari perbandingan model tersebut yaitu didapatkan oleh model YOLOv3 dengan waktu 15ms, sedangkan waktu pemrosesan terlama yaitu model Faster R-CNN dengan kecepatan pemrosesan 2763ms. Penelitian

menggunakan model YOLOv5 dilakukan juga dengan melatih dan memvalidasinya menggunakan dataset gambar mobil yang diambil di jalan iraqi, dimana model YOLOv5 tersebut dapat mengklasifikasi jenis mobil dengan nilai mAP dan skor F1 terendah sebesar 92,4% dan 90%. Model DETR-SPP digunakan untuk mendeteksi kendaraan pada dataset MS COCO dengan hasil mAP 51,31% dan lama waktu pemrosesan 53 FPS [9]. Dengan dataset yang sama pada percobaan tersebut, nilai mAP menggunakan model DETR lebih tinggi dibandingkan YOLOv3-tiny dan faster R-CNN namun memiliki lama waktu pemrosesan lebih lama dibandingkan kedua model tersebut [9].

Penelitian kami yang dibahas pada paper ini memiliki batasan penelitian mengenai model *machine learning* untuk mendeteksi mobil dan klasifikasi jenis mobil dari data gambar yang diambil pada tempat parkir luar gedung TULT Telkom University. Data hasil deteksi mobil dan klasifikasi jenis mobil tersebut dapat berguna untuk pengolahan data selanjutnya seperti deteksi ketersediaan slot parkir mobil, reservasi mobil, dan lainnya. Data gambar kami ambil dari CCTV yang diletakan berhadapan dengan blok parkir mobil yang dilakukan untuk melatih model ML dan validasi performa model ML. Data gambar tersebut kemudian diaugmentasi dengan memotong, melakukan *mosaic*, menambah kecerahan, mengurangi kecerahan, dan membalikan secara horizontal. Lalu model tersebut dilatih dengan mengubah beberapa *hyperparameter* *hyperparameter* model YOLOv8s yaitu *weight decay* dan *freeze*. Nilai *weight decay* yang dicoba yaitu 0.0005, 0.001, 0.01, dan 0.1. Layer yang dilakukan *freeze* dicoba dua kali yaitu *freeze* sebanyak 4 layer dan *freeze* sebanyak 8 layer. Selain itu *hyperparameter* 'mosaic', 'hsv_s', dan 'hsv_v' dimatikan, data gambar telah diaugmentasi di luar pelatihan model YOLOv8s.

II. KAJIAN TEORI

Bagian ini kan menjelaskan teori-teori mengenai model YOLOv8s, Augmentasi Data Gambar, dan *weight decay*.

A. YOLOv8

YOLOv8 adalah salah satu model machine learning YOLO yang dapat mendeteksi objek dan klasifikasi kelas objek secara bersamaan. Arsitektur model YOLOv8 terdiri dari tiga komponen utama yaitu backbone, neck, dan head. Komponen backbone terdiri dari convolutional neural network untuk mengekstrak fitur dari inputan data gambar. Komponen neck digunakan untuk menyatukan feature maps dari beberapa tahapan dari backbone untuk mendapatkan informasi dalam beberapa skala. Head digunakan untuk membuat prediksi mengenai bounding box, nilai dari objek-objek, dan probabilitas dari kelas dari setiap potongan sel. Prediksi-prediksi tersebut kemudian disatukan untuk membuat keputusan terakhir. Tiga inovasi yang terdapat pada model YOLOv8 yaitu spatial attention, feature fusion, dan bottlenecks dan SPPF. Spatial attention digunakan untuk memberi bobot yang lebih tinggi pada fitur yang menonjol pada setiap frame sehingga lokalisasi objek menjadi lebih presisi. Feature fusion dilakukan pada modul C2F, dimana

modul tersebut digunakan untuk menyatukan fitur-fitur semantik dengan informasi spasial yang rendah untuk meningkatkan akurasi pada objek kecil. Bottlenecks dalam CSPDarknet53 backbone digunakan untuk mengurangi kompleksitas komputasi ketika SPPF digunakan untuk menangkap fitur dalam ukuran banyak untuk meningkatkan performa deteksi. Model YOLOv8 yang kami gunakan berjenis YOLOv8s (YOLOv8 small), dimana kedalaman, lebar, dan rasio dari YOLOv8s adalah 0.33, 0.5, dan 2.0.

B. Weight Decay pada YOLOv8

Weight decay pada YOLOv8 menggunakan regularisasi L2 untuk mengatasi masalah *overfitting*. Regularisasi L2 digunakan untuk meminimalisasi fungsi loss dengan membuat bobot menjadi kecil namun tidak menjadikan bobotnya bernilai nol. Regularisasi tersebut memberi penalti pada jumlah kuadrat bobot. Persamaan (1) menunjukan rumus fungsi loss menggunakan regularisasi L2.

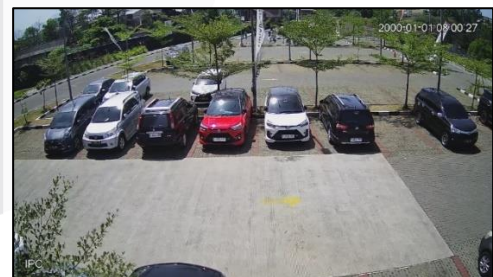
$$\text{Fungsi Loss} = \frac{1}{N} \sum_{i=1}^N (\hat{Y} - Y)^2 + \lambda \sum_{i=1}^N \theta_i^2 \quad (1)$$

III. METODE

Beberapa langkah dilakukan pada penelitian ini yaitu mempersiapkan data, melatih model, validasi model, dan analisis hasil validasi model.

A. Persiapan Data Gambar

Data gambar diambil secara langsung menggunakan kamera CCTV yang diletakan pada tiang bambu, dengan tinggi peletakan kamera 4 meter. Gambar tempat parkir diambil dengan posisi kamera berhadapan dengan barisan slot tempat parkir. Didapatkan 32 gambar dari hasil pengangkapan gambar tempat parkir di luar gedung TULT.



GAMBAR 1
CONTOH HASIL PENANGKAPAN GAMBAR TEMPAT PARKIR DARI CCTV

Beberapa bagian dari gambar tersebut dipotong agar model *machine learning* hanya melatih dan melakukan validasi terhadap slot parkir yang diinginkan (slot parkir yang berhadapan). Gambar dipotong dengan sebatas slot yang diinginkan, agar nilai hasil performa tidak bias atau hasil hasil performa model *machine learning* hanya menunjukan slot yang diinginkan. Gambar dapat dipotong menjadi beberapa bagian, dimana Gambar 2 () menunjukan hasil gambar tempat parkir yang dipotong menjadi dua bagian.



GAMBAR 2
CONTOH HASIL PEMOTONGAN GAMBAR MENJADI DUA BAGIAN

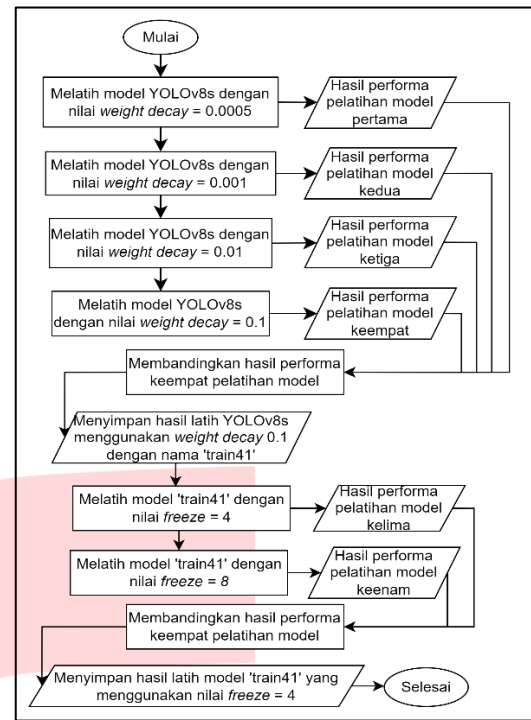
Setelah gambar dipotong, kemudian objek mobil pada gambar tersebut dianotasi dengan empat kelas jenis mobil. Kelas jenis mobil tersebut bernama ‘hatchback’, ‘mpv’, ‘sedan’, dan ‘suv’. Data gambar tersebut dibagi menjadi data latih dan data validasi sebanyak 75% dan 25% secara berurutan. Lalu dilakukan augmentasi sebanyak tiga kali dari jumlah data tanpa augmentasi pada data latih dengan jenis augmentasi *mosaic*, menambah tingkat kecerahan 10%, mengurangi tingkat kecerahan 10%, dan membalikan gambar secara horizontal dengan persentase dilakukan augmentasi setiap jenis augmentasi sebesar 50%. Jumlah kelas jenis ‘hatchback’, ‘mpv’, ‘sedan’, dan ‘suv’ pada data latih sebanyak 549, 451, 70, dan 400 secara berurutan, sedangkan jumlah setiap kelas pada data validasi secara berurutan adalah 30, 21, 3, dan 16 secara berurutan.



GAMBAR 3
(CONTOH DATA GAMBAR SETELAH DIAUGMENTASI)

B. Pelatihan Model YOLOv8s

Model YOLOv8s dilatih dengan mengganti beberapa *hyperparameter* model tersebut. *Hyperparameter* yang diubah secara tetap yaitu Epoch, Batch, *mosaic*, *hsv_s*, dan *hsv_v* dengan nilai 150, 16, 0, 0, dan 0 secara berurutan. Nilai *mosaic*, *hsv_s*, dan *hsv_v* diubah menjadi nol agar tidak terjadi augmentasi kembali, karena data gambar telah kami augmentasi sebelumnya. Lalu model dilatih dengan parameter *weight decay*, dimana empat kali pelatihan model menggunakan *weight decay* yang berbeda-beda dengan nilai 0.0005, 0.001, 0.01, dan 0.1. Setelah mendapatkan performa model YOLOv8s terbaik yaitu dengan melatihnya menggunakan *weight decay* sebesar 0.1, kemudian model hasil pelatihan tersebut disimpan dengan nama model ‘train41’. Lalu model ‘train41’ dilatih kembali dengan menggunakan dataset yang sama, dimana pada pelatihan ini dilakukan dua kali dengan *hyperparameter freeze* yang berbeda beda. *Hyperparameter freeze* digunakan untuk membekukan beberapa *layer* pertama pada model YOLOv8s. Pembekuan empat *layer* dan delapan *layer* model YOLOv8s yang telah dilatih (‘train41’) dilakukan, dimana hasil pembekuan 4 *layer* menghasilkan hasil lebih baik dibandingkan dengan pembekuan 8 *layer*. Alur pelatihan model YOLOv8s ditunjukkan pada gambar 4 (Diagram alur pelatihan model YOLOv8s)



GAMBAR 4
(DIAGRAM ALUR PELATIHAN MODEL YOLOV8S)

C. Validasi Model YOLOv8s

Performa model YOLOv8s untuk mendeteksi objek mobil dan klasifikasi jenis objek mobil yang telah dilatih divalidasi secara otomatis ketika pelatihan model YOLOv8s telah selesai. Nilai-nilai performa yang digunakan untuk dianalisis yaitu *mAP*, *precision*, *recall*, dan *F1-Score*. Nilai-nilai performa tersebut dicari dengan menggunakan rumus masing-masing yang berbeda, dimana rumus-rumus tersebut ditunjukkan pada tabel 1 (Persamaan parameter performa model deteksi objek dan klasifikasi objek pada gambar). Pada table persamaan tersebut terdapat beberapa variabel sebagai komponen untuk menghitung parameter performa model deteksi dan klasifikasi yaitu TP (*True Positive*), FP (*False Positive*), N (jumlah data masing-masing jenis kelas objek mobil), dan n (jumlah kelas jenis mobil).

TABEL 1
(PERSAMAAN PARAMETER PERFORMA MODEL DETEKSI OBJEK DAN KLASIFIKASI OBJEK PADA GAMBAR)

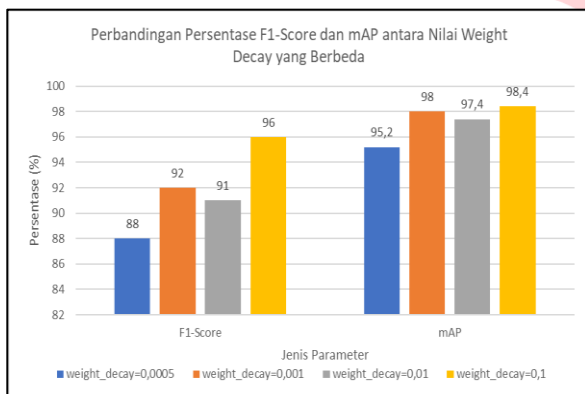
Parameter Performa Model Deteksi dan Klasifikasi	Persamaan
<i>precision</i>	$precision = \frac{TP}{TP + FP}$
<i>recall</i>	$recall = \frac{TP}{TP + FN}$
<i>F1-Score</i>	$F1 = \frac{2 \times precision \times recall}{precision + recall}$
<i>AP (Average Precision)</i>	$AP = \frac{1}{N} \sum_{r=0,0}^{r=1,0} Precision(r)$
<i>mAP (Mean Average Precision)</i>	$mAP = \frac{1}{n} \times \sum_{k=1}^{k=n} AP_k$

IV. HASIL DAN PEMBAHASAN

Hasil parameter performa model YOLOv8s yang telah dilatih dibandingkan hasil parameter performanya antara *hyperparameter* yang berbeda-beda. Perbandingan tersebut dilakukan dari sisi *hyperparameter weight decay* dan *freeze*.

A. Hasil dan Pembahasan Performa Model YOLOv8s yang telah dilatih dengan Nilai *Weight Decay* yang Berbeda

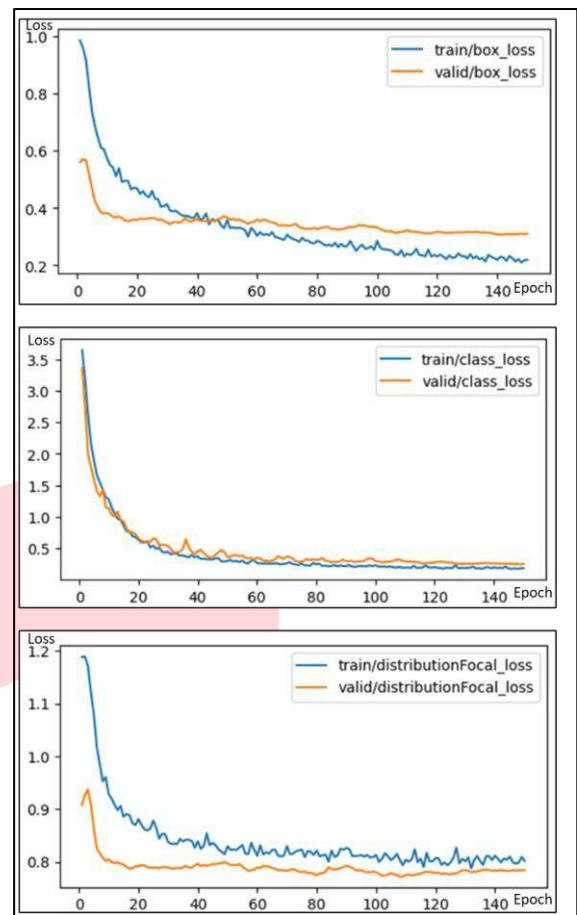
Nilai *weight decay* yang berbeda-beda mempengaruhi performa model YOLOv8s dari segi nilai parameter F1-Score dan mAP. Nilai F1-Score tertinggi didapatkan ketika nilai *weight decay* adalah 0.1 yaitu 96%, sedangkan nilai F1-Score terendah didapatkan ketika model dilatih dengan nilai *weight decay* 0.0005 atau bawaan dari YOLOv8s yaitu 88%. Model yang dilatih *weight decay* 0.001 dan 0.01 memiliki nilai F1-Score 92% dan 91% secara berurutan. Nilai mAP tertinggi didapatkan ketika nilai *weight decay* 0.1 yaitu 98.4%, sedangkan mAP terendah didapatkan ketika nilai *weight decay* 0.0005 yaitu sebesar 95.2%. Nilai mAP ketika nilai *weight decay* 0.001 dan 0.01 yaitu 98% dan 97,4% secara berurutan.



GAMBAR 5

DIAGRAM BATANG PERBANDINGAN PERFORMA MODEL YOLOV8S YAITU F1-SCORE DAN MAP DENGAN NILAI *WEIGHT DECAY* YANG BERBEDA-BEDA

Grafik *box loss* dan *distribution focal loss* yang pada gambar 6, menunjukkan bahwa terdapat sedikit *overfitting* pada model yang dilatih. Pada grafik *box loss* dan *distribution focal loss*, kurva loss pada data latih dan data validasi sedikit berbeda, dimana pada kurva *box loss* nilai *loss* data validasi berada dibawah *loss* data latih, namun ketika *epoch* di atas 50 nilai *loss* data validasi berada di atas *loss* data latih. Nilai *loss* data validasi pun berada dibawah nilai *loss* data latih, namun setelah *epoch* di atas 20, nilai kedua *loss* tersebut mendekat, dimana penurunan signifikan terjadi pada *loss* data latih, namun penurunan tidak signifikan pada *loss* data validasi. Hal tersebut menjadi ciri bahwa model *overfitting* yang disebabkan oleh jumlah data gambar yang kurang.

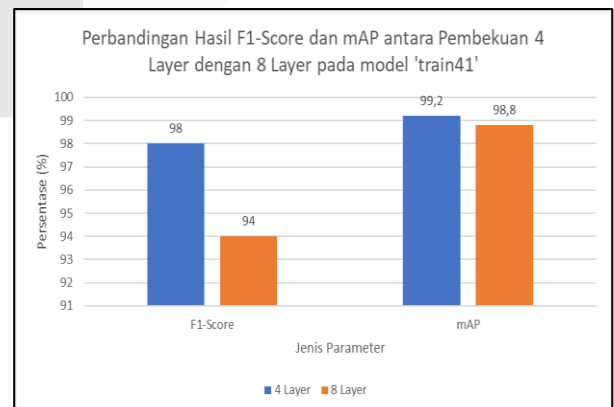


GAMBAR 6

KURVA LOSS MODEL PADA DATA LATIH DAN DATA VALIDASI MENGENAI *BOX LOSS*, *CLASS LOSS*, DAN *DISTRIBUTION FOCAL LOSS*

B. Hasil dan Pembahasan Performa Model YOLOv8s yang telah dilatih dengan Nilai *Freeze* yang Berbeda

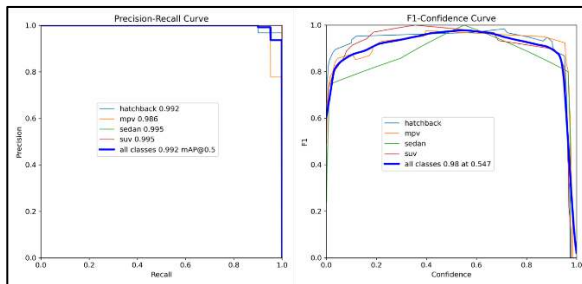
Model dari pelatihan model YOLOv8s dengan menggunakan nilai *weight decay* 0.1 disimpan dengan nama 'train41' lalu dilatih dengan mencoba menggunakan nilai *freeze* 4 layer dan 8 layer. Hasil parameter F1-Score pada *freeze* 4 layer lebih tinggi yaitu 98% dibandingkan dengan *freeze* 8 layer yaitu 94%. Hasil parameter mAP pada *freeze* 4 layer lebih tinggi yaitu 99.2% dibandingkan dengan menggunakan *freeze* 8 layer yaitu sebesar 98.8%.



GAMBAR 7

DIAGRAM BATANG PERBANDINGAN PERFORMA MODEL 'TRAIN41' YAITU F1-SCORE DAN MAP DENGAN NILAI *FREEZE* YANG BERBEDA

Kurva *precision-recall* menunjukkan bahwa model memiliki nilai yang seimbang antara *precision* dan *recall*. Terdapat ketidak seimbangan *precision-recall* pada kelas 'suv', dimana nilai *precision* lebih tinggi dibandingkan nilai *recall*. Kurva rata-rata semua kelas hampir menyentuh bagian sisi bagian kanan dan atas. Pada kurva *F1-confidence* menunjukkan *F1-Score* tertinggi didapatkan ketika nilai *confidence* 0.547. Lama waktu proses model 'train41' dari hasil pelatihan menggunakan *hyperparameter freeze 4 layer* yaitu sebesar 1.95 milidetik.



GAMBAR 8
(KURVA NILAI *PRECISION-RECALL* DAN NILAI *F1-CONFIDENCE*
DARI HASIL *FREEZE 4*)

V. KESIMPULAN

Pada paper ini kami mengajukan model YOLOv8s untuk mendeteksi objek mobil dan mengklasifikasi jenis mobil dari tangkapan gambar CCTV di tempat parkir luar gedung TULT. Nilai parameter performa model berupa *precision*, *recall*, *F1-Score*, dan mAP berada di atas paper lain dari studi literatur yang kami lakukan setelah kami lakukan augmentasi data gambar dan mengubah beberapa *hyperparameter*. Namun, hasil tersebut menunjukkan bahwa model YOLOv8s yang telah kami latih kemungkinan tidak baik pada data baru karena nilai *loss* pada data latih dengan *loss* pada data validasi berbeda cukup jauh. Masih diperlukan pelatihan model tersebut dengan menggunakan jumlah dataset yang lebih banyak, agar model tidak *overfitting*.

REFERENSI

- [1] M. Farda and C. Baliyepalli, "Exploring the Effectiveness of Demand Management Policy in Reducing Traffic Congestion and Environmental

Pollution: Car Free Day and Odd-Even Plate Measures for Bandung City in Indonesia," *Case Studies on Transport Policy*, vol. 6, no. 4, p. 14, 2018.

- [2] C. Biyik, Z. Allam, G. Pieri and D. Moroni, "Smart Parking Systems: Reviewing the Literature, Architecture and Ways Forward," *Smart Cities*, vol. 4, no. 2, pp. 623-642, 2021.
- [3] J. J. Barriga, J. Sulca, A. Ulloa, J. L. Leon, D. Portero, R. Andrade and S. G. Yoo, "Smart Parking: A Literature Review from the Technological Perspective", *Applied Sciences*, vol. 9, no. 21, p. 34, 2019.
- [4] A. Fahim, M. Hasan and M. A. Chowdhury, "Smart parking systems: comprehensive review based on various aspects," *Heliyon*, 2021.
- [5] F. Al-Turjman and A. Malekoo, "Smart parking in IoT-enabled cities: A survey," *Sustainable Cities and Society*, vol. 49, p. 101608, 2019.
- [6] M. V. Peppas, D. Bell, T. Komar and W. Xiao, "URBAN TRAFFIC FLOW ANALYSIS BASED ON DEEP LEARNING CAR DETECTION FROM CCTV IMAGE SERIES," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences/International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vols. XLII-4, pp. 499-506, 2018.
- [7] C. Yang and L. Yuchi, "Object Detection in the KITTI Dataset using YOLO and Faster R-CNN," 2023.
- [8] A. Qazzaz and M. Ahmed, "Car Detection and Features Identification Based on YOLOV5," 2022.
- [9] P. S. Krishendu and P. Mohandas, "DETR-SPP: a fine-tuned vehicle detection with transformer," *Multimedia Tools and Application*, vol. 83, pp. 25573-25594, 2023.