

## Penerapan YOLO dan OpenCV dalam Klasifikasi Kendaraan pada Lalu Lintas Kota Depok

Aldo Satriyo Pamungkas<sup>1</sup>, Shevti Arbekti Arman<sup>2(\*)</sup>, Elliya Sestri<sup>3</sup>

<sup>1, 2, 3</sup>Institut Teknologi dan Bisnis Ahmad Dahlan, Jakarta

### Abstract

*The growth in the number of vehicles in Depok City has driven the need for an accurate and efficient traffic monitoring system. This study implements the You Only Look Once (YOLO) version 8 algorithm to automatically detect and classify vehicles based on Python and OpenCV. The focus of the study is on four types of vehicles, namely motorcycles, private cars, buses, and trucks. The dataset was obtained from CCTV recordings and field documentation, then annotated using LabelImg and processed into YOLO format. The training process was carried out using the pretrained YOLOv8 model, while the system testing was conducted on videos of Depok City roads. Model performance was evaluated using the metrics of mAP@0.5 and mAP@0.5:mAP95, precision, recall, and F1 score. The evaluation results show that the model achieved a mAP@0.5 of 91% and a mAP@0.5:mAP95 of 75.1%, precision of 88.5%, recall of 85.2%, and an F1-score of 86.8%. With these results, the model is capable of detecting and classifying vehicles in real time with high accuracy under various lighting conditions and camera angles. Additionally, this system is integrated with a web interface using Flask for direct visualization of detection results. This research contributes to supporting smart transportation systems in urban environments and provides a potential solution for data-based traffic management.*

**Kata Kunci:** YOLOv8, OpenCV, Python, Vehicle Classification, Object detection.

Informasi Artikel:

Juli – Desember 2025, Vol 6 (2) : hlm 112 – 122

Dikirim : 03 September 2025

Institut Teknologi dan Bisnis Ahmad Dahlan

Diterima : 28 Januari 2026

Diterbitkan : 29 Januari 2026

---

(\*) Korespondensi: [aldopamungkas645@gmail.com](mailto:aldopamungkas645@gmail.com) (Aldo Satriyo Pamungkas)

## PENDAHULUAN

Pertumbuhan jumlah kendaraan di kawasan perkotaan menjadi tantangan besar dalam pengelolaan transportasi modern. Masyarakat cenderung memilih kendaraan pribadi karena faktor kontrol penuh atas waktu perjalanan, fleksibilitas, dan tingkat kenyamanan yang lebih tinggi, sehingga kendaraan pribadi menjadi pilihan utama terutama di kawasan perkotaan (Nureka, 2024). Kondisi ini menyebabkan jumlah kendaraan pribadi terus meningkat dari tahun ke tahun dan berdampak langsung terhadap kemacetan lalu lintas serta tekanan terhadap infrastruktur jalan.

Perkembangan transportasi di Kota Depok turut mengalami peningkatan seiring pertumbuhan jumlah penduduk dan aktivitas ekonomi. Data Badan Pusat Statistik (BPS, 2024) menunjukkan bahwa jumlah kendaraan bermotor di Kota Depok pada tahun 2023 mencapai 1.139.603 unit, terdiri dari 192.821 mobil penumpang, 698 bus, 19.245 truk, dan 926.839 sepeda motor. Peningkatan jumlah kendaraan tersebut mendorong kebutuhan akan sistem pemantauan lalu lintas yang akurat dan efisien. Namun, sistem pemantauan yang ada masih didominasi oleh kamera CCTV yang hanya berfungsi sebagai tampilan visual dan memerlukan pengawasan manual, sehingga rentan terhadap human error dan tidak mampu menyediakan data real time yang berkelanjutan (berita.depok.go.id, 2023).

Dalam pengelolaan lalu lintas modern, deteksi dan klasifikasi kendaraan menjadi aspek penting karena memungkinkan identifikasi serta pengelompokan kendaraan berdasarkan jenis seperti sepeda motor, mobil, bus, dan truk. Informasi tersebut bermanfaat dalam perencanaan transportasi, optimasi infrastruktur jalan, hingga analisis dampak lingkungan. Oleh sebab itu, pengembangan sistem berbasis kecerdasan buatan (Artificial Intelligence/AI) yang mampu melakukan analisis secara otomatis dan real time menjadi kebutuhan mendesak (Sadik, et al., 2024). Salah satu metode yang banyak digunakan dalam bidang computer vision adalah object detection. Algoritma You Only Look Once (YOLO) menjadi solusi populer karena mampu mendeteksi objek secara real time dengan tingkat akurasi tinggi (AI\_Pioneer, 2023). Versi terbaru, YOLOv8, memiliki peningkatan signifikan dibandingkan versi sebelumnya dengan pendekatan anchor-free yang lebih efisien.

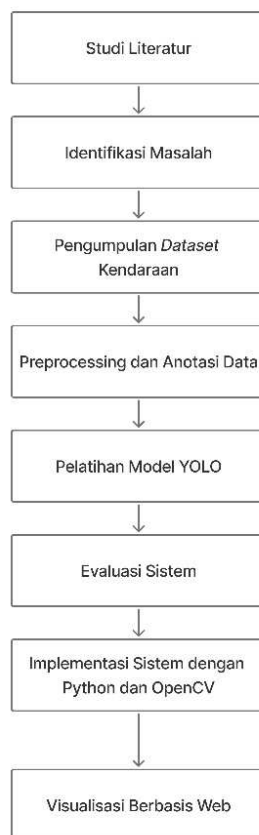
Sejumlah penelitian sebelumnya telah memanfaatkan YOLO untuk mendeteksi dan mengklasifikasikan kendaraan. Mulyana dan Rofik (2022) mengembangkan sistem deteksi real time menggunakan YOLOv5 dan mencapai akurasi 90% untuk berbagai jenis kendaraan. Nugroho, et al. (2024) menggabungkan YOLOv4 dengan OpenCV untuk mendeteksi kendaraan secara real time dari video, sedangkan Ramadhani, et al. (2024) menggunakan YOLO untuk identifikasi kendaraan bermotor pada rekaman dashcam. Pratama, et al. (2023) membuktikan bahwa YOLOv7 mampu melakukan klasifikasi kendaraan dengan akurasi 86–91%. Penelitian lain oleh Telaumbanua, et al. (2023) menegaskan efektivitas YOLOv8 dalam deteksi kendaraan di konteks Indonesia. Namun, penelitian yang mengimplementasikan YOLOv8 secara langsung khususnya di wilayah Kota Depok dengan integrasi real time menggunakan OpenCV masih terbatas.

Berdasarkan kesenjangan penelitian tersebut, penelitian ini dilakukan untuk mengembangkan sistem deteksi dan klasifikasi kendaraan otomatis berbasis YOLOv8, Python, dan OpenCV yang diuji pada rekaman lalu lintas di Kota Depok. Dengan adanya sistem ini, diharapkan dapat mendukung efisiensi dan akurasi pengelolaan transportasi, sekaligus menjadi dasar pengambilan keputusan yang lebih tepat dalam menghadapi kompleksitas lalu lintas perkotaan.

Tujuan penelitian ini adalah untuk mengimplementasikan model YOLOv8 dalam klasifikasi kendaraan berbasis OpenCV Python di wilayah Kota Depok, sehingga dapat mendukung peningkatan sistem pemantauan lalu lintas secara otomatis, akurat, dan real time.

## METODE

Metodologi penelitian ini menjelaskan bagaimana penelitian dilakukan secara sistematis mulai dari perancangan, pengumpulan data, pengolahan, hingga analisis hasil. Penelitian dirancang untuk mengembangkan sistem deteksi dan klasifikasi kendaraan berbasis algoritma YOLO dengan memanfaatkan bahasa pemrograman Python dan pustaka OpenCV. Desain penelitian meliputi studi literatur, identifikasi masalah, pengumpulan data, preprocessing, pelatihan model, evaluasi, implementasi sistem, hingga visualisasi berbasis web.



Gambar 1. Kerangka Kerja Penelitian

- a. Penelitian ini diawali dengan studi literatur melalui penelaahan buku, jurnal, dokumentasi, dan penelitian terdahulu yang berkaitan dengan kecerdasan buatan, computer vision, deteksi objek, algoritma YOLO, dan pustaka OpenCV. Hasil kajian ini menjadi dasar dalam pemilihan metode, perangkat, serta arsitektur sistem yang digunakan.
- b. Selanjutnya dilakukan identifikasi masalah dengan menyoroti keterbatasan sistem pemantauan lalu lintas di Kota Depok yang masih bersifat manual melalui kamera CCTV. Dari masalah tersebut ditetapkan tujuan penelitian, yaitu membangun sistem otomatis berbasis AI untuk deteksi dan klasifikasi kendaraan secara real time.
- c. Lokasi penelitian ditetapkan di Jalan Ir. H. Juanda menuju Jl. Saminten, Depok, Jawa Barat, yang merupakan jalur padat lalu lintas. Waktu penelitian dilaksanakan secara bertahap, dimulai dari pengumpulan data, pelatihan model, hingga implementasi sistem.
- d. Tahap berikutnya adalah pengumpulan data, berupa foto kendaraan dari hasil dokumentasi lapangan di Kota Depok serta data tambahan dari sumber daring seperti roboflow, pinterest, dan CCTV publik. Dokumentasi lapangan dilakukan selama tiga minggu dengan variasi sudut, jarak, dan pencahayaan. Data citra digunakan sebagai dataset pelatihan, sementara data video dimanfaatkan untuk pengujian sistem.
- e. Data yang diperoleh diproses melalui preprocessing dan anotasi. Gambar yang blur atau tidak relevan dihapus, kemudian setiap kendaraan (motor, mobil, bus, truk) diberi bounding box menggunakan LabelImg. Dataset diekspor dalam format YOLO (.txt) sehingga siap digunakan untuk pelatihan, berikut jumlah data yang telah di anotasi:

Tabel 1. Data Anotasi

<b>Jumlah Gambar</b>	<b>Training</b>	<b>Validation</b>	<b>Objek Anotasi</b>
7464	5980	1484	13.663

- f. Pelatihan model dilakukan dengan Python dan GPU. Parameter utama yang digunakan antara lain jumlah kelas = 4, epoch = 150, batch size = 16, dan learning rate = 0.001. Kinerja model dipantau melalui loss, precision, recall, dan mAP.
- g. Tahap evaluasi sistem dilakukan menggunakan confusion matrix, precision, recall, mAP, dan F1-score. Evaluasi bertujuan menilai akurasi model dalam kondisi nyata dan mengidentifikasi kemungkinan kesalahan klasifikasi.
- h. Setelah itu, dilakukan implementasi sistem dengan mengintegrasikan model YOLO ke dalam aplikasi Python berbasis OpenCV. Sistem membaca input video secara real time, mendeteksi kendaraan, menampilkan bounding box serta label, dan menyimpan hasil video deteksi.

Sebagai tahap akhir, dibangun visualisasi berbasis web menggunakan Flask. Antarmuka ini menampilkan hasil deteksi secara interaktif sehingga dapat diakses dengan mudah untuk keperluan presentasi, demonstrasi, dan evaluasi.

## HASIL DAN PEMBAHASAN

### a. Implementasi Sistem

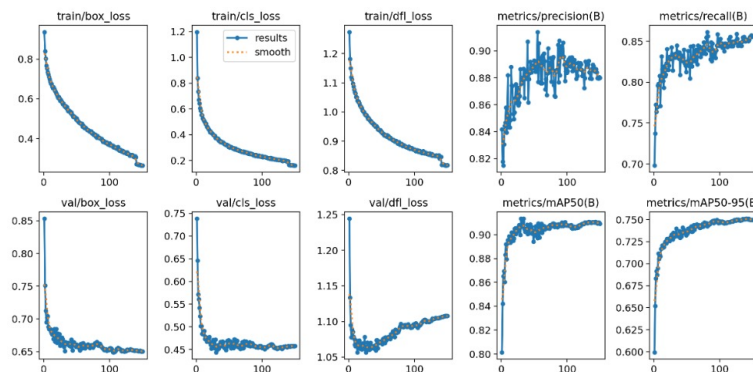
Implementasi sistem dilakukan setelah seluruh tahapan perancangan dan pelatihan model YOLOv8 berhasil diselesaikan. Sistem ini dirancang menggunakan bahasa pemrograman Python dengan beberapa library pendukung untuk proses deteksi, klasifikasi, pelacakan objek, visualisasi, serta penyimpanan hasil ke database. Sistem diimplementasikan dalam bentuk aplikasi berbasis web menggunakan Flask, dan mendukung input berupa video rekaman maupun live stream CCTV.



Gambar 2. Implementasi Sistem YOLOv8

### b. Hasil Evaluasi Model YOLOv8

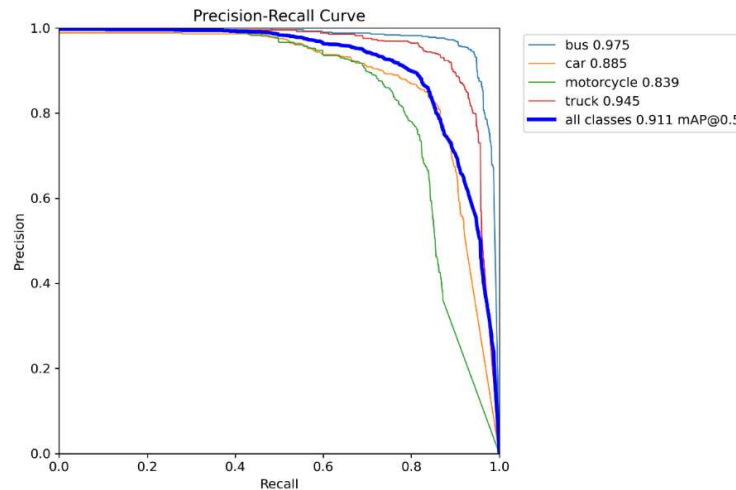
Evaluasi kinerja model YOLOv8 dilakukan untuk menilai performa deteksi dan klasifikasi objek kendaraan (bus, car, motorcycle, dan truck) setelah pelatihan selama 150 epoch. Hasil evaluasi menunjukkan tren yang positif dan stabil. Visualisasi grafik nilai loss (box loss, class loss, dan DFL) menunjukkan penurunan yang konsisten pada data pelatihan dan validasi. Hal ini mengindikasikan bahwa model belajar dengan baik dan memiliki kemampuan generalisasi yang stabil, sehingga dapat meminimalkan risiko overfitting. Seiring dengan itu, metrik evaluasi seperti precision, recall, dan Mean Average Precision (mAP) mengalami peningkatan bertahap, menegaskan perbaikan performa model.



Gambar 3. Evaluasi model

c. Hasil Evaluasi. **mAP@0.5**

Evaluasi mAP (Mean Average precision) merupakan salah satu indikator utama dalam menilai performa sistem deteksi objek. Pada gambar Precision-recall Curve, ditampilkan kurva precision terhadap recall untuk masing-masing kelas objek. Grafik ini menggambarkan seberapa baik model dapat melakukan prediksi yang akurat (precision) dan lengkap (recall) terhadap objek yang ada. Berikut gambar evaluasi precision dan *recall curve*:



Gambar 4. Hasil Model mAP@0.5

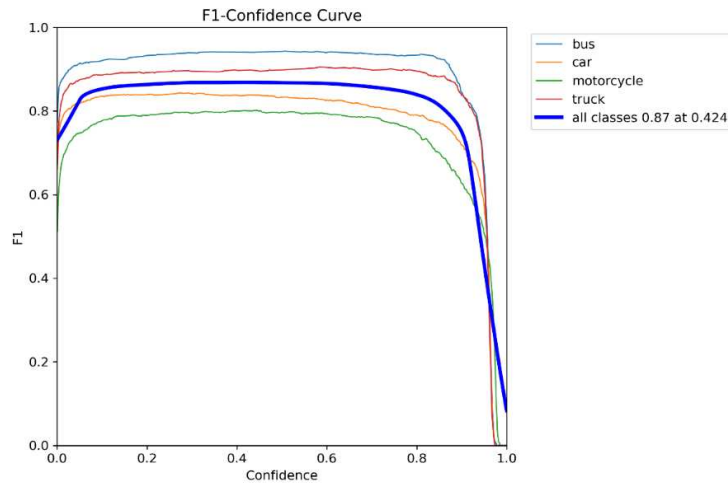
Interpretasi nilai **mAP@0.5** (rata-rata presisi pada ambang *Intersection over Union* 0.5) untuk setiap kelas adalah sebagai berikut:

1. **Bus:** 0.975 (sangat tinggi)
2. **Truk:** 0.945 (sangat baik)
3. **Mobil (Car):** 0.885 (baik)
4. **Sepeda motor (Motorcycle):** 0.839 (terendah)

Secara keseluruhan, rata-rata mAP@0.5 dari seluruh kelas adalah **0.911**, menunjukkan akurasi model yang tinggi. Performa terbaik terlihat pada objek yang relatif besar, seperti bus dan truk, yang lebih mudah dikenali oleh model dibandingkan objek kecil seperti sepeda motor.

d. Hasil Evaluasi F1 *Confidence Score*

Evaluasi F1 *Score* dilakukan untuk memberikan gambaran yang seimbang antara *precision* dan *recall* dalam satu metrik tunggal. Grafik F1-*Confidence* Curve menggambarkan hubungan antara nilai confidence threshold dan skor F1 untuk masing-masing kelas.



Gambar 5. Hasil F1 Score

Hasil analisis menunjukkan bahwa nilai F1 tertinggi secara keseluruhan adalah 0.87, yang dicapai pada confidence threshold 0.424. Nilai ini dapat dijadikan ambang batas (*threshold*) default untuk implementasi sistem pendeteksi objek, karena mampu menyeimbangkan prediksi positif yang benar (*true positives*) dan menghindari kesalahan deteksi (*false positives dan false negatives*).

Secara spesifik, kelas bus dan truk menunjukkan performa yang paling stabil dan tinggi. Sebaliknya, sepeda motor menunjukkan kurva F1 yang lebih rendah dan tidak stabil, yang mengindikasikan tantangan dalam deteksi objek kecil dan kompleks. Secara umum, kurva F1 membuktikan stabilitas model dalam menghasilkan prediksi yang akurat di berbagai tingkat kepercayaan.

- e. Evaluasi dilakukan setelah pelatihan selama 150 epoch untuk menilai performa deteksi kendaraan (*bus, car, motorcycle, truck*). Hasil evaluasi model ditampilkan dalam bentuk tabel metrik utama berikut:

Tabel 2. Hasil Evaluasi Model YOLOv8

Kelas	Precision	Recall	mAP@0.5	F1-Score
Bus	0.97	0.96	0.975	0.93
Truck	0.94	0.95	0.945	0.91
Car	0.89	0.88	0.885	0.86
Motorcycle	0.84	0.83	0.839	0.80
<b>Rata-rata</b>	<b>0.91</b>	<b>0.90</b>	<b>0.911</b>	<b>0.87</b>

Berdasarkan tabel tersebut, terlihat bahwa kelas **bus dan truck** memiliki kinerja terbaik dengan precision dan recall di atas 0.94, sedangkan **motorcycle** memperoleh nilai terendah karena ukurannya kecil dan pergerakannya cepat sehingga lebih sulit dikenali. Rata-rata mAP@0.5 sebesar **0.911** menunjukkan model memiliki akurasi tinggi dalam mendeteksi dan mengklasifikasikan kendaraan

- f. Hasil Pengujian Visualisasi Berbasis Web  
Pengujian dilakukan secara lokal menggunakan server Flask dan database MongoDB dengan input berupa rekaman CCTV. Seluruh fungsi sistem, termasuk web dashboard, diuji melalui antarmuka browser. Sebelum memulai pengujian, server diaktifkan melalui terminal dengan langkah-langkah:

1. Mengaktifkan environment virtual.
2. Masuk ke direktori aplikasi.
3. Menjalankan berkas Flask (app.py).

Hasil pengujian menunjukkan bahwa semua komponen utama sistem berfungsi dengan baik. Aplikasi **Flask** berhasil memproses dan menyajikan data deteksi secara waktu-nyata (*real time*). Sementara itu, **MongoDB** sukses menyimpan data kendaraan secara dinamis. Antarmuka web yang informatif membuktikan sistem dapat diakses dan digunakan dengan efektif.

Secara keseluruhan, sistem ini berhasil sebagai prototipe efisien untuk pemantauan kendaraan berbasis video dan memiliki potensi untuk dikembangkan lebih lanjut dalam aplikasi berskala besar seperti pemantauan lalu lintas cerdas (*smart traffic monitoring*) atau sistem parkir otomatis.

g. Hasil Pengujian Perhitungan Video

Pengujian sistem deteksi dan pelacakan kendaraan dilakukan menggunakan dua video berdurasi tiga menit yang direkam di titik putaran Saminten, Kota Depok. Tujuan pengujian adalah untuk mengukur kemampuan sistem dalam menghitung jumlah kendaraan berdasarkan jenis dan arah pergerakannya.

Perbedaan hasil antara Pengujian 1 dan Pengujian 2 disebabkan oleh perbedaan kondisi lalu lintas saat perekaman. Pengujian 1 dilakukan saat volume kendaraan lebih padat, sehingga menghasilkan jumlah deteksi yang lebih tinggi dibandingkan Pengujian 2, yang dilakukan saat lalu lintas lebih sepi.

Tabel 3. Jumlah Deteksi Kendaraan

Jenis Kendaraan	Pengujian 1	Pengujian 2
<i>Car</i>	64	24
<i>Bus</i>	0	1
<i>Motorcycle</i>	155	82
<i>Truck</i>	23	13

Tabel 4. Arah Pergerakan Kendaraan

Arah Crossing	Pengujian 1	Pengujian 2
Masuk	144	52
Keluar	98	68

Sistem deteksi dan pelacakan kendaraan dengan YOLOv8 dan ByteTrack mampu mengidentifikasi jenis serta arah kendaraan dengan baik. Motorcycle paling dominan terdeteksi, disusul car, truck, dan bus yang hanya sekali terdeteksi. Beberapa kendaraan mungkin terlewat akibat hambatan visual, sudut kamera, atau kualitas video rendah, namun secara keseluruhan sistem tetap efisien dan akurat.

## KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa implementasi model YOLOv8 untuk klasifikasi dan pelacakan kendaraan berbasis OpenCV dan Flask telah berhasil dilakukan dengan efektif. Model ini menunjukkan

performa yang sangat baik dalam mendeteksi dan mengklasifikasikan empat jenis kendaraan, yaitu mobil, motor, bus, dan truk.

Tingkat akurasi model terbukti tinggi, sebagaimana ditunjukkan oleh nilai metrik evaluasi yang sangat baik pada data validasi. Nilai mAP@0.5 mencapai 0.910, sementara F1-Score menunjukkan angka 86.8%, menegaskan kemampuan model untuk memberikan prediksi yang akurat dan seimbang.

Sistem ini tidak hanya unggul dalam akurasi model, tetapi juga dalam fungsionalitasnya. Model YOLOv8 berhasil diintegrasikan ke dalam sistem berbasis web menggunakan Flask yang memungkinkan deteksi dan visualisasi data secara real-time. Lebih lanjut, dengan menggabungkan algoritma ByteTrack dan basis data MongoDB, sistem ini mampu melacak kendaraan, menghitung jumlah kendaraan masuk dan keluar secara otomatis, serta menyimpan data tersebut dengan efisien. Secara keseluruhan, sistem yang dikembangkan ini efektif sebagai solusi prototipe untuk pemantauan lalu lintas berbasis video.

## DAFTAR PUSTAKA

- Adi Nugroho, S., Kahfi, M., Fidzri Akhbar Alamsyah, M., Natanael, A., & Rosyani, P. (2024). Studi Kasus Penggunaan YOLO dan OpenCV untuk MENDETEKSI JENIS KENDARAAN di JALAN. *Jurnal Artificial Inteligent Dan Sistem Penunjang Keputusan*, 2(2). <https://jurnalmahasiswa.com/index.php/aidanspk>
- AI\_Pioneer. (2023, June 28). *Object detection with YOLO and OpenCV: A Practical Guide* | by AI\_Pioneer | Medium. <https://medium.com/%40tejasdalvi927/object-detection-with-yolo-and-opencv-a-practical-guide-cf7773481d11>
- Badan Pusat Statistik (BPS). (2024). Jumlah Kendaraan Bermotor Menurut Kabupaten/Kota dan Jenis Kendaraan di Provinsi Jawa Barat (unit), 2023 - Tabel Statistik - Badan Pusat Statistik Provinsi Jawa Barat. <https://jabar.bps.go.id/id/statistics-table/3/VjJ3NGRGa3dkRk5MTIU1bVNFOTVVbmQyVURSTVFUMDkjMw==/jumlah-kendaraan-bermotor-menurut-kabupaten-kota-dan-jenis-kendaraan-di-provinsi-jawa-barat--unit---2023.html?year=2023>
- berita.depok.go.id. (2023). Wali Kota Depok Pantau Lalu Lintas Kota Lewat 125 CCTV. <https://Berita.Depok.Go.Id/Wali-Kota-Depok-Pantau-Lalu-Lintas-Kota-Lewat-125-Cctv>.
- Chavan, C., Hembade, S., Jadhav, G., Komalwad, P., & Rawat, P. (2023). *Computer vision Application Analysis based on Object detection*. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, 07(04). <https://doi.org/10.55041/ijsrem19015>
- Ciksadan, C., Soim, S., & Jami, N. (2024). Desain dan Pengembangan *Website* untuk Mendeteksi Malware Menggunakan Framework *Flask* yang Diintegrasikan dengan *Machine learning*. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 7(3), 1213–1218. <https://doi.org/10.32493/jtsi.v7i3.42003>
- Dr. Basuki Rahmat, S. S. M., & Budi Nugroho, S. K. M. K. (2021). *Pemrograman Deep Learning dengan Python*.
- Dr. Budi Raharjo, S. Kom. , M. Kom. , MM. (2021). *SISTEM MANAJEMEN DATABASE*.

- Dr. Arie Gunawan. (2024). Mobile Programming Menggunakan Flutter dan Visual Studio Code Untuk Pemula. [www.penerbitlitnus.co.id](http://www.penerbitlitnus.co.id)
- ella siman. (2023). What Is a *Dataset*? Definitive Guide. <https://Brightdata.Com/Blog/Web-Data/What-Is-a-Dataset>. <https://brightdata.com/blog/web-data/what-is-a-dataset>
- Gede, I., Sudipa, I., & Darmawiguna, M. (2024). BUKU AJAR DATA MINING. <https://www.researchgate.net/publication/377415198>
- Hashmi, K. A., Pagani, A., Stricker, D., & Afzal, M. Z. (2022). BoxMask: Revisiting *Bounding box* Supervision for Video *Object detection*. <http://arxiv.org/abs/2210.06008>
- Hussain, M. (2023). YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. In *Machines* (Vol. 11, Issue 7). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/machines11070677>
- Interaction Design Foundation. (2023). *User Interface (UI) Design*. <https://www.interaction-design.org/literature/topics/ui-design>
- Joseph Teguh Santoso. (2023). KECERDASAN BUATAN.
- labelImg. (n.d.). <https://PyPi.Org/Project/LabelImg/>.
- Lintang, H., Hafidz, F., Rafi, M., Naufal, A., & Muthalib, M. A. (2024). Penerapan AI untuk Optimasi Rute Secara *Real time* dan Meningkatkan Efisiensi Pengiriman. *Jurnal Sains Masyarakat*, 01. <https://doi.org/10.1016/j.ijlm.2022.04.003>
- Malik, U. (2022). Image Processing in Open CV. *International Journal for Research in Applied Science and Engineering Technology*, 10(6), 2664–2666. <https://doi.org/10.22214/ijraset.2022.44527>
- Marpaung, F., Aulia, F., Suryani SKom, N., & Cyra Nabila SKom, R. (2022). *COMPUTER VISION DAN PENGOLAHAN CITRA DIGITAL*. [www.pustakaaksara.co.id](http://www.pustakaaksara.co.id)
- Melanie. (2023). What is a *dataset*? How do I work with it? <https://Datascientest.Com/En/What-Is-a-Dataset-How-Do-i-Work-with-It>. <https://datascientest.com/en/what-is-a-dataset-how-do-i-work-with-it>
- Mulyana, D. I., & Rofik, M. A. (2022). Implementasi Deteksi *Real time* Klasifikasi Jenis Kendaraan Di Indonesia Menggunakan Metode YOLOV5.
- Muttaqin, Muhammad Munsarif, Green Ferry Mandias, Wahyu Wijaya Widiyanto, Stenly Richard Pungus, Agung Widarman, Suryani, Aslam Fatkhudin, Eva Firdayanti Bisono, Pasnur, Nurirwan Saputra, Siska Aprilia Hardiyanti, Mochammad Anshori, Eva Firdayanti Bisono, & Wiranti Kusuma Hapsari. (2024). Pengenalan Data Mining.
- opencv. (n.d.). [opencv](https://Opencv.Org/About/). <https://Opencv.Org/About/>.
- Raharjo, B. (2021). Pembelajaran Mesin (*Machine learning*).
- Ramadhani, F., Satria, A., & Dewi, S. (2024). Identifikasi Kendaraan Bermotor pada Dashcam Mobil Menggunakan Algoritma YOLO. *Hello World Jurnal Ilmu Komputer*, 2(4), 199–206. <https://doi.org/10.56211/helloworld.v2i4.466>
- Sadik, N., Hossain, T., & Sayeed, F. (2024). *Real time* Detection and Analysis of Vehicles and Pedestrians using Deep Learning.
- Saha, S. (2024). Traffic Monitoring System Using *Machine learning* And Python OpenCV and YOLOv8. <https://doi.org/10.13140/RG.2.2.12182.46408>
- Soebroto, A. A. (2019). Buku Ajar AI, *Machine learning* & Deep Learning. <https://www.researchgate.net/publication/348003841>
- Stuart J. Russel, & Peter Norvig. (2021). *Artificial Intelligence A Modern Approach Fourth Edition*.

- Szeliski, R. (2021). *Computer vision: Algorithms and Applications* 2nd Edition. <https://szeliski.org/Book>,
- Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A Comprehensive Review of YOLO Architectures in *Computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS*. In *Machine learning and Knowledge Extraction* (Vol. 5, Issue 4, pp. 1680–1716). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/make5040083>
- Ujjwal Sharma , Tanya Goel , Dr. Jagbeer Singh. (2023). *Real time Image Processing Using Deep Learning With Opencv And Python*. Journal of Pharmaceutical Negative Results, 1905–1908. <https://doi.org/10.47750/pnr.2023.14.03.246>
- What is Python? Executive Summary | Python.org. (n.d.). Retrieved May 6, 2025, from <https://www.python.org/doc/essays/blurb/>
- W3C. (2024). HTML *Living standard*. World Wide Web Consortium. Retrieved from <https://html.spec.whatwg.org/multipage/>
- Mozilla Developer Network. (2023). CSS: Cascading Style Sheets. Mozilla Foundation. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/CSS>
- Mozilla Developer Network. (2024). JavaScript Guide. Mozilla Foundation. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
- Hui, J. (2019). mAP (*Mean Average precision*) for *Object detection*. Retrieved from <https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>
- V7 Labs. (2023). What is *Mean Average precision* (mAP) and How it Works. Retrieved from <https://www.v7labs.com/blog/mean-average-precision>
- Albon, C. (2020). *Machine learning with Python for Everyone*. Addison-Wesley Professional.
- Jupyter Project. (2021). *Teaching and Learning with Jupyter*. Retrieved from <https://jupyter4edu.github.io/jupyter-edu-book/>
- SuperAnnotate. (2024). Understanding mAP in *Object detection*. Retrieved from <https://www.superannotate.com/blog/mean-average-precision-and-its-uses-in-object-detection>
- Pratama, B. A., Rahman, S., & Sembiring, A. (2023). *Klasifikasi Jenis Kendaraan pada Jalan Raya Menggunakan YOLOv7*. Jurnal Informatika Teknologi dan Sains (JINTEKS), 5(4). Model YOLOv7 berhasil melakukan klasifikasi jenis kendaraan seperti mobil, sepeda motor, bus, dan truk dengan akurasi mencapai 86% untuk video dan 91% untuk gambar.
- Telaumbanua, A. P. H., Larosa, T. P., Pratama, P. D., Fauza, R. H., & Husein, A. M. (2023). *Vehicle Detection and Identification Using Computer vision Technology with the Utilization of the YOLOv8 Deep Learning Method*. Sinkron: Jurnal dan Penelitian Teknik Informatika, 7(4), 2150–2157. Penelitian ini menunjukkan kemampuan YOLOv8 dalam mendeteksi dan mengidentifikasi kendaraan secara akurat di Indonesia.