



Perancangan Dan Pengembangan Aplikasi Deteksi Objek Pergerakan Mobil Di Jalan Tol Menggunakan Opencv Dan Python

Hafizd Ramadhan¹, Bagas Daniswara Adi Pangestu², Elvarel Naufal Krisna³

^{1,2,3} Sistem Informasi, Universitas Pamulang

¹hafizdramadhan031@gmail.com, ²bagasdanis1104@gmail.com, ³elfarel.krisna82@gmail.com

Abstrak

Penelitian ini mengembangkan sistem deteksi pergerakan mobil berbasis OpenCV dan Python untuk mendukung pemantauan lalu lintas secara waktu nyata (*real-time*). Metode yang digunakan meliputi prapemrosesan citra, deteksi objek dengan algoritma *background subtraction* (MOG2 dan KNN), pelacakan menggunakan *centroid tracking*, serta klasifikasi kondisi lalu lintas menjadi tiga kategori: lancar, padat, dan macet. Evaluasi kinerja dilakukan berdasarkan akurasi deteksi, *frame per second* (FPS), dan pengujian berdasarkan pencahayaan. Hasil pengujian menunjukkan bahwa sistem mampu mencapai akurasi 94% pada siang hari, 88% pada malam hari, dan 80% pada kondisi hujan, dengan rata-rata FPS berkisar antara 18–32 tergantung resolusi video. Temuan ini membuktikan bahwa pendekatan berbasis OpenCV dapat diimplementasikan secara efektif untuk mendukung sistem transportasi cerdas dan pemantauan lalu lintas yang efisien.

Kata Kunci: Deteksi Objek, Opencv, Pemantauan Lalu Lintas

1. Pendahuluan

Perkembangan teknologi transportasi cerdas (*Intelligent Transportation System* atau ITS) telah mendorong pemanfaatan teknologi pengolahan citra dan *computer vision* dalam mendukung pengelolaan lalu lintas secara lebih efektif. Pertumbuhan jumlah kendaraan yang terus meningkat menyebabkan berbagai permasalahan seperti kemacetan, keterlambatan perjalanan, peningkatan konsumsi bahan bakar, serta tingginya risiko kecelakaan lalu lintas. Oleh karena itu, diperlukan sistem pemantauan lalu lintas yang mampu memberikan informasi kondisi jalan secara cepat dan akurat sebagai dasar pengambilan keputusan dalam manajemen transportasi.

Salah satu teknologi yang banyak digunakan dalam pemantauan lalu lintas adalah deteksi objek berbasis video. Teknologi ini memungkinkan sistem untuk mengenali, melacak, dan menghitung jumlah kendaraan yang melintas secara otomatis tanpa memerlukan intervensi manusia. Pemanfaatan kamera CCTV yang telah banyak terpasang di jalan tol memberikan peluang besar untuk mengembangkan sistem pemantauan lalu lintas berbasis *computer vision* yang mampu bekerja secara *real-time*.

OpenCV (*Open Source Computer Vision Library*) merupakan pustaka pengolahan citra yang banyak digunakan dalam pengembangan aplikasi deteksi objek karena menyediakan berbagai fungsi untuk akuisisi citra, pemrosesan video, segmentasi objek, serta visualisasi hasil deteksi. Selain itu, bahasa pemrograman Python dipilih karena memiliki sintaks yang sederhana, mudah dikembangkan, serta didukung oleh berbagai pustaka kecerdasan buatan dan pembelajaran mesin yang dapat meningkatkan kemampuan sistem dalam mengenali objek kendaraan.

Beberapa penelitian sebelumnya telah menerapkan metode deteksi kendaraan menggunakan teknik *background subtraction*, Haar Cascade, maupun pendekatan berbasis *deep learning*. Meskipun mampu menghasilkan tingkat akurasi yang cukup baik, sebagian penelitian masih berfokus pada proses deteksi objek tanpa melakukan analisis kondisi lalu lintas secara menyeluruh. Selain itu, pengujian pada kondisi lingkungan yang berbeda seperti siang hari, malam hari, dan cuaca hujan masih relatif terbatas sehingga diperlukan penelitian lanjutan untuk mengetahui stabilitas kinerja sistem dalam berbagai kondisi operasional.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi deteksi objek pergerakan mobil di jalan tol menggunakan OpenCV dan Python. Sistem yang dikembangkan mampu mendeteksi kendaraan secara *real-time*, menghitung jumlah kendaraan yang melintas, serta mengklasifikasikan kondisi lalu lintas ke dalam kategori lancar, ramai lancar, dan macet. Dengan adanya sistem ini diharapkan proses pemantauan lalu lintas dapat dilakukan secara lebih efektif, cepat, dan akurat sehingga dapat mendukung penerapan sistem transportasi cerdas di Indonesia.

Kontribusi utama penelitian ini adalah implementasi teknologi OpenCV dan Python untuk mendeteksi pergerakan kendaraan secara *real-time*, pengembangan mekanisme klasifikasi kondisi lalu lintas berdasarkan jumlah kendaraan yang terdeteksi, serta evaluasi kinerja sistem pada berbagai kondisi lingkungan guna mengetahui tingkat akurasi dan keandalan sistem yang dikembangkan.

2. Metode Penelitian

Penelitian ini menggunakan metode pengembangan sistem dengan pendekatan **System Development Life Cycle (SDLC)** model Waterfall. Metode ini dipilih karena memiliki tahapan yang sistematis dan terstruktur sehingga sesuai untuk pengembangan aplikasi deteksi objek kendaraan berbasis computer vision. Tahapan penelitian meliputi analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan evaluasi kinerja sistem.

2.1 Analisis Kebutuhan

Pada tahap ini dilakukan identifikasi kebutuhan perangkat keras dan perangkat lunak yang digunakan dalam pengembangan sistem. Perangkat keras terdiri dari komputer atau laptop yang digunakan untuk pemrosesan data serta sumber video berupa rekaman CCTV jalan tol. Adapun perangkat lunak yang digunakan meliputi sistem operasi Windows 10, Python 3.10, OpenCV, Ultralytics YOLOv8, NumPy, dan Roboflow untuk pengelolaan dataset.

Data penelitian berupa video lalu lintas jalan tol yang diperoleh dari kamera pengawas (CCTV). Video tersebut digunakan sebagai sumber data untuk proses deteksi dan penghitungan kendaraan secara real-time.

2.2 Pengolahan Data

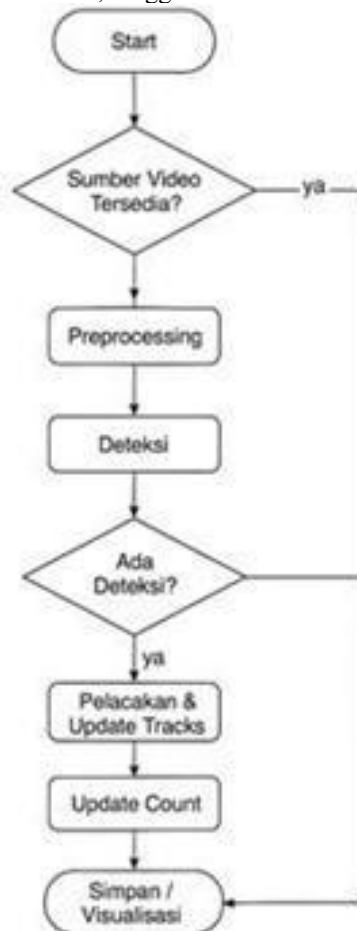
Data video yang diperoleh selanjutnya diekstraksi menjadi frame-frame citra untuk memudahkan proses analisis. Tahap pengolahan data meliputi:

1. Akuisisi video lalu lintas.
2. Ekstraksi frame video.
3. Anotasi objek kendaraan menggunakan bounding box.
4. Augmentasi data berupa rotasi, flipping, dan penyesuaian kontras.
5. Pembagian dataset menjadi data pelatihan (92%), data validasi (5%), dan data pengujian (3%).

Tahapan ini bertujuan untuk meningkatkan kualitas data sehingga model mampu mengenali objek kendaraan pada berbagai kondisi lingkungan.

2.3 Perancangan Sistem

Sistem dirancang untuk mendeteksi kendaraan yang melintas pada jalan tol secara real-time. Alur kerja sistem dimulai dari penerimaan input video, preprocessing citra, deteksi kendaraan menggunakan model YOLOv8, pelacakan objek, perhitungan jumlah kendaraan, hingga klasifikasi kondisi lalu lintas.



Gambar 1. Flowchart

Kondisi lalu lintas diklasifikasikan berdasarkan jumlah kendaraan yang terdeteksi pada setiap frame, yaitu:

DOI: <https://doi.org/10.69693/ijmst.v4i2.10936>

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

- Lancar : 1–5 kendaraan
- Ramai Lancar : 6–15 kendaraan
- Macet : >15 kendaraan

Informasi hasil deteksi ditampilkan dalam bentuk bounding box, label objek, nilai confidence, jumlah kendaraan, dan status kondisi lalu lintas.

2.4 Implementasi Model YOLOv8

Model YOLOv8 digunakan sebagai algoritma utama untuk mendeteksi kendaraan pada setiap frame video. Model dilatih menggunakan dataset yang telah dianotasi dan diproses sebelumnya. Selama proses inferensi, setiap objek kendaraan yang terdeteksi akan menghasilkan koordinat bounding box, kelas objek, dan nilai confidence.

Hasil deteksi kemudian diteruskan ke modul pelacakan objek untuk mempertahankan identitas kendaraan pada setiap frame sehingga proses penghitungan kendaraan dapat dilakukan secara akurat.

2.5 Pengujian dan Evaluasi Sistem

Pengujian sistem dilakukan menggunakan metode black-box testing untuk memastikan seluruh fungsi sistem berjalan sesuai kebutuhan. Selain itu, dilakukan evaluasi performa model menggunakan beberapa metrik pengukuran, yaitu Accuracy, Precision, Recall, Mean Average Precision (mAP), dan Frame Per Second (FPS).

Akurasi deteksi kendaraan dihitung menggunakan Persamaan (1).

$$\text{Accuracy} = \frac{N_{\text{detected}}}{N_{\text{total}}} \times 100\%$$

Keterangan:

- (N_{detected}) = jumlah kendaraan yang berhasil dideteksi dengan benar.
- (N_{total}) = jumlah kendaraan sebenarnya pada video pengujian.

Pengujian dilakukan pada tiga kondisi lingkungan yang berbeda, yaitu siang hari, malam hari, dan hujan. Hasil pengujian digunakan untuk menganalisis pengaruh kondisi pencahayaan dan cuaca terhadap kinerja sistem deteksi kendaraan.

2.6 Evaluasi Hasil

Data hasil pengujian dianalisis secara deskriptif untuk membandingkan tingkat akurasi deteksi dan kecepatan pemrosesan sistem pada masing-masing kondisi pengujian. Analisis dilakukan untuk mengetahui kemampuan sistem dalam mendukung pemantauan lalu lintas secara real-time serta potensi implementasinya pada Intelligent Transportation System (ITS)

3. Hasil dan Pembahasan

Pengembangan sistem deteksi kendaraan berbasis YOLOv8 dilakukan melalui beberapa tahapan utama yang meliputi pengumpulan dan pembuatan dataset, proses anotasi data, pelatihan model, evaluasi performa model, serta implementasi sistem untuk mendeteksi dan memantau kendaraan secara real-time. Setiap tahapan dilakukan secara sistematis untuk memastikan model yang dihasilkan mampu mendeteksi kendaraan dengan tingkat akurasi yang baik pada berbagai kondisi lingkungan. Hasil dari setiap tahapan tersebut kemudian dianalisis dan dibahas untuk mengetahui kinerja sistem yang dikembangkan. sebagai berikut:

1. Pengolahan Data dan Dataset

Data diperoleh dari rekaman video lalu lintas atau CCTV yang kemudian diproses menggunakan platform Roboflow. Tahapan pengolahan data meliputi anotasi objek kendaraan menggunakan bounding box, augmentasi data seperti rotasi, flipping, dan penyesuaian kontras, serta konversi dataset agar kompatibel dengan framework deep learning. Dataset yang telah diproses kemudian dibagi menjadi tiga bagian, yaitu data pelatihan (*train set*) sebesar 92%, data validasi (*validation set*) sebesar 5%, dan data pengujian (*test set*) sebesar 3%.

2. Pelatihan Model YOLOv8

Model YOLOv8 dilatih menggunakan dataset hasil preprocessing dan anotasi yang telah disiapkan. Proses pelatihan dilakukan untuk menghasilkan model yang mampu mengenali objek kendaraan pada berbagai kondisi lingkungan. Teknik augmentasi data digunakan untuk meningkatkan kemampuan generalisasi model terhadap variasi sudut pengambilan gambar, pencahayaan, dan kondisi lalu lintas. Kinerja model dievaluasi menggunakan metrik *Precision*, *Recall*, dan *Mean Average Precision (mAP)* untuk mengukur tingkat ketepatan deteksi kendaraan.

3. Modul Sistem

Sistem yang dikembangkan terdiri atas beberapa modul utama yang saling terintegrasi, yaitu:

- Modul Akuisisi Data**, berfungsi untuk menerima input berupa video atau kamera secara real-time.
- Modul Preprocessing**, berfungsi untuk melakukan penyesuaian resolusi, peningkatan kualitas citra, dan pengurangan noise sebelum proses deteksi dilakukan.

- c. **Modul Deteksi Kendaraan**, berfungsi untuk mendeteksi kendaraan menggunakan model YOLOv8 berdasarkan nilai *confidence threshold* yang telah ditentukan.
- d. **Modul Tracking dan Perhitungan Kendaraan**, berfungsi untuk mempertahankan identitas objek pada setiap frame serta menghitung jumlah kendaraan yang melintas pada area pengamatan.
- e. **Modul Visualisasi dan Penyimpanan**, berfungsi untuk menampilkan *bounding box*, label kendaraan, jumlah kendaraan, serta kondisi lalu lintas secara real-time dan menyimpan hasil pengolahan data.

Implementasi sistem merupakan tahap realisasi rancangan menjadi aplikasi yang dapat berjalan secara nyata. Pada tahap ini, model deteksi YOLOv8, modul pelacakan, perhitungan jumlah kendaraan, serta klasifikasi kondisi lalu lintas diintegrasikan dalam satu sistem yang mampu menerima input dari video maupun kamera secara real-time. Hasil implementasi menunjukkan bahwa sistem mampu menampilkan visualisasi kondisi lalu lintas dalam tiga situasi berbeda:

1. Lancar



Gambar 1 Tampilan Sistem Saat Lancar

Pada kondisi ini sistem mendeteksi dua kendaraan yang terdiri dari mobil dan truk. Setiap kendaraan ditampilkan dalam bentuk *bounding box* yang dilengkapi label objek dan nilai *confidence*. Karena jumlah kendaraan yang terdeteksi berada pada rentang 1–5 kendaraan, sistem mengklasifikasikan kondisi lalu lintas sebagai **Lancar**.

2. Ramai Lancar



Gambar 2 Tampilan Sistem Saat Ramai Lancar

Pada kondisi ini sistem mendeteksi sejumlah kendaraan yang terdiri dari mobil, truk, dan bus. Kepadatan lalu lintas mengalami peningkatan dibandingkan kondisi sebelumnya, namun arus kendaraan masih dapat bergerak dengan normal. Berdasarkan jumlah kendaraan yang terdeteksi, sistem menampilkan status **Ramai Lancar** sebagai indikator kondisi lalu lintas.

3. Macet



Gambar 3 Tampilan Sistem Saat Macet

Pada kondisi ini sistem mendeteksi hingga 26 kendaraan yang memenuhi area jalan. *Bounding box* kendaraan terlihat saling berdekatan yang menunjukkan tingginya kepadatan lalu lintas. Karena jumlah kendaraan melebihi batas kategori yang telah ditentukan, sistem mengklasifikasikan kondisi lalu lintas sebagai **Macet** dan menampilkan indikator visual untuk memudahkan pengguna dalam mengidentifikasi kondisi tersebut.

Secara keseluruhan, sistem mampu mendeteksi dan mengidentifikasi kendaraan, menampilkan posisi objek menggunakan *bounding box*, menghitung jumlah kendaraan yang melintas, serta mengklasifikasikan kondisi lalu lintas secara real-time. Hasil visualisasi menunjukkan bahwa sistem dapat memberikan informasi kepadatan lalu lintas secara jelas dan berpotensi diterapkan sebagai pendukung sistem pemantauan transportasi berbasis *Intelligent Transportation System (ITS)*.

Tabel 1. Pengujian Berdasarkan Kondisi Pencahayaan

Kondisi Uji	Jumlah Mobil	Mobil Terdeteksi	Akurasi	FPS
Siang cerah	50	48	96%	25
Malam	50	38	76%	23
Hujan	50	30	60%	20

Berdasarkan hasil pengujian pada Tabel 1, sistem menunjukkan performa terbaik pada kondisi siang hari dengan tingkat akurasi sebesar 96% dan kecepatan pemrosesan rata-rata 25 FPS. Kondisi pencahayaan yang optimal memungkinkan model YOLOv8 mendeteksi kendaraan dengan lebih akurat sehingga jumlah kendaraan yang tidak terdeteksi relatif sedikit.

Pada kondisi malam hari, akurasi sistem menurun menjadi 76% dengan kecepatan pemrosesan rata-rata 23 FPS. Penurunan performa ini disebabkan oleh rendahnya intensitas cahaya yang memengaruhi kualitas citra, sehingga beberapa kendaraan tidak terdeteksi atau mengalami kesalahan klasifikasi.

Sementara itu, pada kondisi hujan, sistem memperoleh akurasi sebesar 60% dengan rata-rata kecepatan pemrosesan 20 FPS. Kondisi cuaca yang kurang mendukung menyebabkan kualitas citra menurun akibat gangguan visual seperti percikan air dan berkurangnya visibilitas objek. Hasil pengujian menunjukkan bahwa faktor pencahayaan dan kondisi cuaca memiliki pengaruh yang signifikan terhadap performa sistem deteksi kendaraan berbasis YOLOv8.

Pengujian ini dilakukan untuk mengevaluasi akurasi dan konsistensi sistem dalam mendeteksi jumlah mobil pada setiap frame, sekaligus melihat pengaruh kepadatan objek terhadap performa deteksi. Hasil pengujian ditunjukkan pada tabel berikut:

Tabel 2. Pengujian Berdasarkan Jumlah Mobil dalam Frame

Jumlah Mobil dalam Frame	Akurasi	FPS
1-5	100%	26
6-10	92%	24
>10	88%	21

Berdasarkan Tabel 2, sistem memperoleh akurasi tertinggi sebesar 100% pada kondisi 1–5 kendaraan dalam satu frame dengan rata-rata kecepatan pemrosesan 26 FPS. Hasil tersebut menunjukkan bahwa model mampu mendeteksi seluruh kendaraan dengan baik ketika jumlah objek yang diamati relatif sedikit.

Ketika jumlah kendaraan meningkat menjadi 6–10 kendaraan dalam satu frame, akurasi menurun menjadi 92% dengan rata-rata kecepatan pemrosesan 24 FPS. Penurunan ini terjadi karena sistem harus memproses lebih banyak objek secara bersamaan sehingga kompleksitas deteksi meningkat.

Pada kondisi dengan jumlah kendaraan lebih dari 10 kendaraan dalam satu frame, akurasi sistem turun menjadi 88% dengan rata-rata kecepatan pemrosesan 21 FPS. Kepadatan kendaraan yang tinggi menyebabkan terjadinya tumpang tindih (*occlusion*) antarobjek sehingga proses deteksi menjadi lebih sulit. Selain itu, peningkatan jumlah objek juga menambah beban komputasi yang berdampak pada penurunan nilai FPS.

Secara keseluruhan, hasil pengujian menunjukkan bahwa peningkatan jumlah kendaraan dalam frame berpengaruh terhadap performa sistem. Meskipun demikian, model YOLOv8 masih mampu mempertahankan tingkat akurasi yang relatif tinggi dan kecepatan pemrosesan yang memadai untuk kebutuhan pemantauan lalu lintas secara real-time.

4. Kesimpulan

Penelitian ini berhasil mengembangkan aplikasi deteksi objek pergerakan mobil di jalan tol menggunakan OpenCV, Python, dan model YOLOv8 untuk mendukung pemantauan lalu lintas secara real-time. Sistem yang dibangun mampu mendeteksi kendaraan, menghitung jumlah kendaraan yang melintas, serta mengklasifikasikan kondisi lalu lintas ke dalam kategori lancar, ramai lancar, dan macet berdasarkan tingkat kepadatan kendaraan yang terdeteksi. Hasil pengujian menunjukkan bahwa sistem memperoleh tingkat akurasi sebesar 96% pada kondisi siang hari, 76% pada kondisi malam hari, dan 60% pada kondisi hujan. Selain itu, sistem mampu mempertahankan kecepatan pemrosesan antara 20–25 frame per second (FPS) sehingga memenuhi kebutuhan aplikasi pemantauan lalu lintas secara real-time.

Berdasarkan hasil tersebut, dapat disimpulkan bahwa penerapan OpenCV dan YOLOv8 mampu memberikan performa yang baik dalam mendeteksi kendaraan pada lingkungan jalan tol. Sistem yang dikembangkan berpotensi diterapkan sebagai bagian dari Intelligent Transportation System (ITS) untuk membantu pengelolaan lalu lintas, pemantauan kepadatan kendaraan, serta pengambilan keputusan secara cepat dan berbasis data. Meskipun demikian, performa sistem masih dipengaruhi oleh kondisi pencahayaan dan cuaca, terutama pada malam hari dan saat hujan. Oleh karena itu, penelitian selanjutnya dapat difokuskan pada peningkatan akurasi deteksi melalui penambahan dataset yang lebih beragam, penerapan teknik augmentasi yang lebih kompleks, penggunaan model deteksi yang lebih adaptif terhadap perubahan lingkungan, serta integrasi teknologi edge computing untuk meningkatkan efisiensi pemrosesan pada implementasi skala nyata.

Reference

- Ardianto, D., & Widiyatmoko, A. T. (2024). Color Detector In An Image Using Python And Computer Vision Library. *Journal Of Intelligent Systems And Information Technology*, 1(1), 25–30.
- Budiana, F. P. (2024). *Model Atensi Citra Wajah Audiens Berbasis Edge Machine Learning*. Universitas Islam Sultan Agung.
- Editya, A. S., Ahmad, T., & Studiawan, H. (2025). *Deep Learning & Optical Flow Dalam Analisis Forensik Drone*. Penerbit Andi.
- Fiqri, A., Hugo, A., & Kalbuana, N. (2024). Analisis Penggunaan Drone Untuk Meningkatkan Respons Cepat Dalam Penanganan Kecelakaan Pesawat Di Area Terpencil. *Jurnal Riset Ilmu Kesehatan Umum Dan Farmasi (JRIKUF)*, 2(3), 76–94.
- Fitriyati Prisunia, S. (2023). *Pemanfaatan Jetson Nano Nvidia Untuk Mendeteksi Penggunaan Masker Secara Real-Time Menggunakan Opencv Python* (Skripsi). Universitas Islam Sultan Agung.
- Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing* (4th Ed.). Pearson Education.
- Kaehler, A., & Bradski, G. (2017). *Learning Opencv 3: Computer Vision In C++ With The Opencv Library*. O'Reilly Media.
- Khulan, M. A., & Pebrianti, R. (2023). *Rancang Bangun Sistem Keamanan Parkir Kendaraan Roda Dua Dengan Teknologi Plate Recognition* (Skripsi). Politeknik Negeri Ujung Pandang.
- Kushariyadi, K., Apriyanto, H., Herdiana, Y., Asy'ari, F. H., Judijanto, L., Prasun, Y. P., & Mardikawati, B. (2024). *Artificial Intelligence: Dinamika Perkembangan AI Beserta Penerapannya*. PT Sonpedia Publishing Indonesia.

- Lukman, A. (2024). *Implementasi Deeplabv3+ Untuk Peningkatan Deteksi Dan Tracking Lajur Jalan Pada Sistem Autonomous Car* (Skripsi). Universitas Hasanuddin.
- Priandini, J. R. (2024). *Pengenalan Rambu Lalu Lintas Menggunakan Model You Only Look Once (YOLO) V8* (Skripsi). Universitas Islam Sultan Agung Semarang.
- Putra, P. Y., Arifianto, A. S., Fitri, Z. E., & Puspitasari, T. D. (2023). Deteksi Kendaraan Truk Pada Video Menggunakan Metode Tiny-Yolov4. *Jurnal Informatika Polinema*, 9(2), 215–222.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition*, 779–788.
- Redmon, J., & Farhadi, A. (2018). Yolov3: An Incremental Improvement. *Arxiv Preprint Arxiv:1804.02767*.
- Suryadi, D., Octiva, C. S., Fajri, T. I., Nuryanto, U. W., & Hakim, M. L. (2024). Optimasi Kinerja Sistem Iot Menggunakan Teknik Edge Computing. *Jurnal Minfo Polgan*, 13(2), 1456–1461.
- Szeliski, R. (2022). *Computer Vision: Algorithms And Applications* (2nd Ed.). Springer.
- Ultralytics. (2024). *Ultralytics Yolov8 Documentation*. Ultralytics. <https://docs.ultralytics.com>
- Yaseen, M. (2024). What Is Yolov8: An In-Depth Exploration Of The Internal Features Of The Next-Generation Object Detector. *Arxiv Preprint Arxiv:2408.15857*.