Jurnal MOSFET
Vol. xx No. xx, September 2025, pp. xx-xx, eISSN: 2775-5274
http://jurnal.umpar.ac.id/index.php/jmosfet
DOI: https://doi.org/10.31850/jmosfet.vxix.xxxx

Sistem Pendeteksi Nomor Plat Kendaraan Menggunakan YOLO Dan Tesseract OCR

Muh. Syakir Muin[1*], Muhammad Basri[2], A. Irmayani Pawelloi[3]

*[1,*2,3]Program Studi Teknik Elektro, Universitas Muhammadiyah Parepare, Indonesia*
**Email : syakirmuin00@gmail.com*

Abstract: Security and automated gate systems improve efficiency and safety in vehicle access control. This research presents the design and implementation of a verified vehicle license plate detection system to automate a portal gate. The system uses You Only Look Once (YOLO) for object detection and Tesseract OCR for character recognition. A Raspberry Pi 4 serves as the processing unit, equipped with a webcam as an input sensor and an MG996R servo motor for gate actuation. In operation, the camera captures a vehicle image, YOLO detects the license plate region, Tesseract OCR reads the characters, and the system verifies the plate against a database. If verified, the Raspberry Pi sends a signal to open the gate. Experiments under varying lighting and angles show the system can accurately detect and recognize license plates, opening the gate only for authorized vehicles. This approach demonstrates high real-time performance and holds promise for parking, residential, and industrial security applications.

Keywords: *License Plate Detection, YOLO, Tesseract OCR, Raspberry Pi 4, Automated Gate Access*

1. PENDAHULUAN

Perkembangan *computer vision* dan kecerdasan buatan telah memungkinkan penerapan sistem keamanan dan kontrol akses yang otomatis pada kawasan parkir dan area terpadu (Smitha *et al.*, 2023). Misalnya, sistem otomatisasi gerbang dapat menggunakan kamera untuk mengenali nomor plat kendaraan secara real-time, menggantikan pemeriksaan manual (Moussaoui *et al.*, 2024). *YOLO (You Only Look Once)* menjadi algoritma deteksi objek populer karena kecepatannya dalam pengolahan gambar secara langsung. Setelah plat terdeteksi, *OCR (Optical Character Recognition)* seperti *Tesseract* dapat mengekstrak karakter dari citra plat tersebut. Menunjukkan bahwa kombinasi platform *Raspberry Pi* dengan algoritma visi komputer dapat menghasilkan deteksi plat nomor secara *near*-real-time dengan biaya rendah. Dengan karakteristik *Raspberry Pi* yang terjangkau dan portabel, sistem ini cocok untuk aplikasi keamanan skala kecil hingga menengah.

Penelitian terdahulu telah membuktikan keandalan pendekatan *YOLO* dan *Tesseract* untuk *Automated License Plate Recognition (ALPR)* – misalnya menggunakan *YOLOv5* untuk deteksi plat dan Tesseract untuk pengenalan karakter. Hasil pengujian menunjukkan bahwa kombinasi ini memiliki akurasi deteksi tinggi dan keandalan baik dalam berbagai kondisi (Thapliyal *et al.*, 2023). Bahkan melaporkan akurasi deteksi 99% dan pengenalan karakter 98% menggunakan *YOLOv8* dan teknik pemrosesan citra tambahan. Urgensi penelitian ini muncul dari kebutuhan masyarakat akan solusi

keamanan terintegrasi yang efisien untuk akses kendaraan, misalnya pada area parkir otomatis atau komplek perumahan. Berdasarkan hal tersebut, penelitian ini bertujuan merancang dan mengimplementasikan sistem pendeteksi nomor plat kendaraan berbasis *YOLO* dan *Tesseract OCR* yang dijalankan pada *Raspberry Pi 4*, sehingga hanya kendaraan terverifikasi yang dapat membuka portal otomatis. Tujuan utama penelitian adalah menguji efektivitas metode *YOLO v8* dan *Tesseract OCR* dalam mendeteksi dan mengenali plat nomor secara real-time dalam kondisi pencahayaan dan sudut pengambilan gambar yang bervariasi.

2.METODOLOGI PENELITIAN

2.1 Jenis Penelitian

Penelitian ini bersifat eksperimental dan pengembangan sistem. Metode yang digunakan mencakup implementasi model deep learning dan pengujian kinerja sistem pada perangkat keras yang sesungguhnya. Metode eksperimental dilakukan dengan menguji model *YOLO* untuk mendeteksi plat nomor dan Tesseract untuk membaca karakternya. Citra kendaraan direkam melalui kamera, lalu YOLO mengenali lokasi plat nomor dan memotong region plat (ROI). ROI tersebut selanjutnya diproses oleh *Tesseract OCR* untuk mengenali setiap karakter.

2.2 Tempat dan Waktu Penelitian

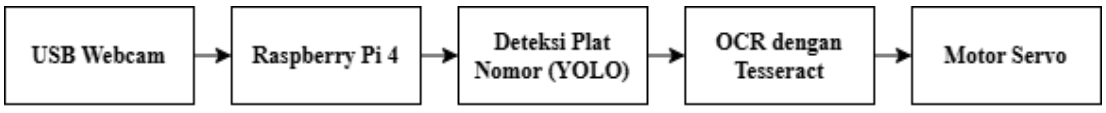
Penelitian dilaksanakan di Laboratorium Teknik Elektro, Universitas Muhammadiyah Parepare. Jadwal penelitian mencakup studi pustaka, perancangan alat dan perangkat lunak, pemasangan perangkat keras, dan pengujian sistem. Uji sistem dilakukan pada bulan terakhir penelitian.

2.3 Alat dan Bahan Penelitian

Perangkat keras utama yang dipakai meliputi Raspberry Pi 4 (single-board computer), kamera webcam USB, motor servo MG996R untuk aktuasi gerbang, serta catu daya dan perangkat pendukung lainnya. Perangkat lunak yang digunakan antara lain OpenCV untuk praproses citra, pustaka *ultralytics* YOLOv8 dalam bahasa Python untuk deteksi objek, dan Tesseract OCR versi terbaru untuk pengenalan teks. Basis data berisi daftar plat nomor terverifikasi disimpan secara lokal di Raspberry Pi.

2.4 Rancangan Penelitian

Blok diagram sistem keseluruhan yang dirancang pada penelitian ini menggambarkan hubungan antar komponen utama ditunjukkan pada Gambar 1.



Gambar 1. Diagram Blok

Pada Gambar 1. Diagram Blok, Pertama, kamera menangkap gambar frame kendaraan secara terus-menerus. Gambar diubah ke grayscale dan dibersihkan dari noise untuk meningkatkan kualitas deteksi. *YOLOv8* kemudian mendeteksi area plat nomor dalam gambar tersebut. Setelah kotak pembatas plat ditemukan, citra ROI plat nomor dipotong dan diproses lebih lanjut untuk memperjelas karakter. Selanjutnya, modul *Tesseract OCR* membaca karakter pada plat dan mengeluarkan string teks. Hasil pembacaan dibandingkan dengan daftar plat terverifikasi. Jika plat cocok, *Raspberry Pi* mengirim sinyal PWM ke motor servo untuk membuka gerbang.

2.5 Teknik Pengumpulan Data

Data pengujian dikumpulkan dengan merekam citra kendaraan dalam berbagai kondisi: pencahayaan siang dan malam, variasi jarak 50 cm dan 80 cm, dan sudut pengambilan 15° dan 30°. Tujuan pengujian ini adalah mengevaluasi tingkat deteksi plat dan akurasi pembacaan teks oleh *OCR* dalam situasi nyata. Kinerja sistem diukur berdasarkan metrik seperti *mean average precision* (mAP) *YOLO* untuk deteksi objek, tingkat keberhasilan deteksi plat, dan akurasi pembacaan karakter. Waktu respon end-to-end dari deteksi hingga aktuasi gerbang juga diukur. Analisis data meliputi evaluasi reliabilitas pada kondisi tertentu. Beberapa studi serupa menunjukkan bahwa arsitektur terdistribusi (misalnya *Raspberry Pi* sebagai *edge* dan server pengolah) dapat menjaga performa waktu-nyata meski sumber daya terbatas

3. HASIL DAN PEMBAHASAN

3.1 Perancangan Perangkat Keras

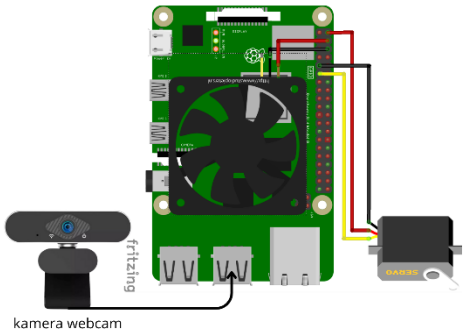
Pada perancangan perangkat keras sistem pendeteksi nomor plat kendaraan ini dirancang untuk mengintegrasikan *Raspberry Pi 4* sebagai unit pemrosesan utama, kamera USB/webcam sebagai sensor pengambil citra, dan motor servo MG996R sebagai aktuator pintu portal. Rangkaian keseluruhan menghubungkan kamera ke port USB 3.0 Raspberry Pi untuk menjaga kestabilan frame rate, sementara servo dikendalikan melalui pin GPIO dengan sinyal PWM.

Raspberry Pi 4 dipilih karena memiliki prosesor *quad-core ARM Cortex-A72 1.5 GHz* dan RAM hingga 4 GB yang mampu menjalankan algoritma *YOLOv8* dan *Tesseract OCR* secara langsung tanpa bantuan komputer eksternal. Kamera diletakkan sejajar dengan tinggi plat kendaraan untuk mengurangi distorsi, dan diuji pada resolusi 640×480 piksel serta 30 FPS agar sesuai dengan kapasitas komputasi Raspberry Pi.

Motor servo MG996R digunakan untuk menggerakkan mekanisme palang portal. Dengan torsi tinggi (±11 kg·cm), servo ini mampu membuka dan menutup palang dengan cepat dan presisi. Servo menerima sinyal PWM dari *Raspberry Pi*, di mana posisi sudut dikontrol untuk membuka dan menutup sesuai status verifikasi plat nomor.

Hasil integrasi menunjukkan bahwa Raspberry Pi, kamera, dan servo dapat bekerja sinkron dengan waktu respon rata-rata 1,5–2 detik dari pengambilan citra hingga

portal terbuka. Hal ini membuktikan bahwa konfigurasi perangkat keras sederhana ini cukup efektif untuk sistem *real-time* dengan kebutuhan energi rendah.

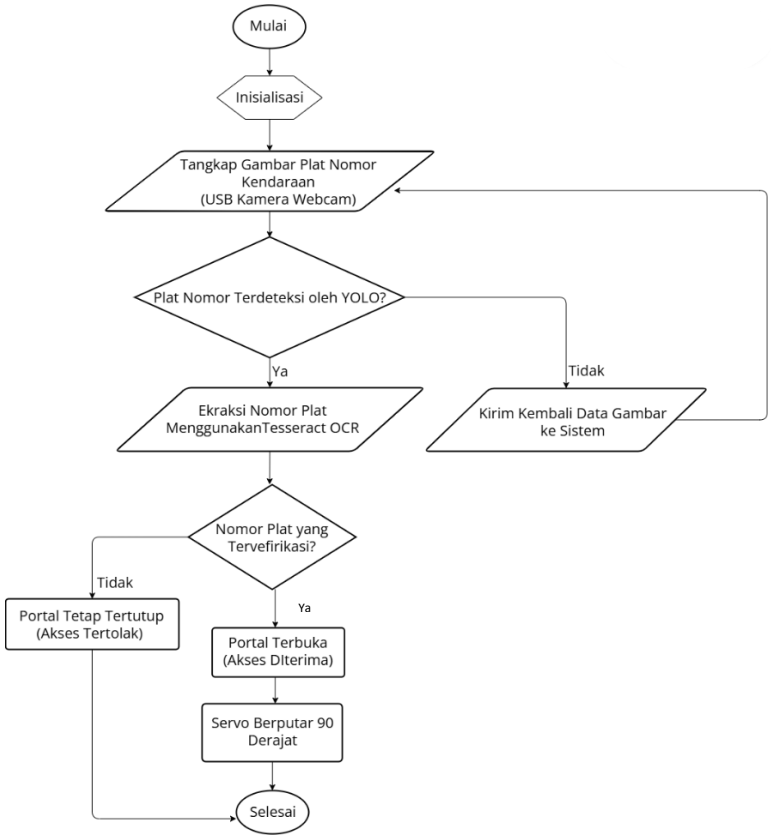


Gambar 2. Rangkaian keseluruhan

3.2 Perancangan Perangkat Lunak

Perangkat lunak dikembangkan menggunakan bahasa *Python* pada sistem operasi Raspberry Pi OS. Tiga komponen utama yang diintegrasikan adalah *YOLOv8* untuk deteksi plat, *OpenCV* untuk praproses citra, dan *Tesseract OCR* untuk pengenalan karakter.

a. Diagram Alir (Flowchart)



Gambar 3. Diagram Alir (Flowchart)

Gambar 3 menunjukkan Alur program dimulai dengan kamera yang menangkap citra kendaraan secara *real-time*. Frame kemudian diproses oleh *YOLOv8* untuk mendeteksi posisi plat nomor dan menghasilkan koordinat *bounding box*. Bagian *ROI (Region of Interest)* yang berisi plat dipotong, lalu melalui tahap praproses meliputi konversi *grayscale*, filtering untuk mengurangi noise, dan *thresholding adaptif* untuk memperjelas kontras karakter.

Hasil praproses kemudian dimasukkan ke modul *Tesseract OCR* untuk mengenali karakter alfanumerik pada plat. Output berupa teks nomor plat selanjutnya diverifikasi dengan basis data plat nomor yang tersimpan di *Raspberry Pi*. Jika cocok, *Raspberry Pi* mengirimkan sinyal ke motor servo untuk membuka portal.

3.3 Hasil Pengujian

Pada tahap ini dilakukan serangkaian pengujian terhadap sistem pendeteksi nomor plat kendaraan menggunakan *YOLO* sebagai pendeteksi objek dan *Tesseract OCR* sebagai pembaca teks. Pengujian dilakukan dengan menggunakan beberapa jenis warna plat nomor kendaraan, variasi kondisi lingkungan (siang dan malam), variasi jarak (50 cm dan 80cm), serta variasi sudut pengambilan gambar (15° dan 30°). Tujuan dari pengujian ini adalah untuk mengevaluasi tingkat keberhasilan deteksi plat, akurasi hasil pembacaan teks *OCR*, serta keterhubungan dengan status portal otomatis yang dikendalikan berdasarkan hasil dari verifikasi nomor plat.

Tabel 1. Klasifikasi plat nomor yang digunakan

Jenis Plat Nomor Kendaraan	Status
	Terverifikasi
	Terverifikasi
	Terverifikasi
	Terverifikasi
	Terverifikasi

Tabel 2. Hasil pengujian plat nomor kuning DP 2724 AY

No	Pengujian	Hasil Deteksi *YOLO*	Hasil OCR (Tesseract)	Status Portal
1	Siang, jarak 50 cm, sudut 15°	Plat terdeteksi	m214AY	Portal tidak terbuka
2	Siang, jarak 50 cm, sudut 30°	Plat terdeteksi	mp2124AY	Portal tidak terbuka
3	Siang, jarak 80 cm, sudut 15°	Plat terdeteksi	mp27gAY	Portal tidak terbuka
4	Siang, jarak 80 cm, sudut 30°	Plat terdeteksi	2104AY	Portal tidak terbuka
5	Malam, jarak 50 cm, sudut 15°	Plat terdeteksi	mp2724AY	Portal tidak terbuka
6	Malam, jarak 50 cm, sudut 30°	Plat terdeteksi	w2124AY	Portal tidak terbuka
7	Malam, jarak 80 cm, sudut 15°	Plat terdeteksi	MP2724ay	Portal tidak terbuka
8	Malam, jarak 80 cm, sudut 30°	Plat terdeteksi	m2124AY	Portal tidak terbuka

Berdasarkan hasil pengujian plat nomor kuning dengan verifikasi nomor plat DP 2724 AY. Pengujian dilakukan terhadap sistem deteksi nomor plat kendaraan dengan kondisi pencahayaan berbeda (siang dan malam), variasi jarak (50 cm dan 80 cm), serta variasi sudut (15° dan 30°). Tabel hasil pengujian pada plat nomor kuning DP 2724 AY menunjukkan bahwa sistem *YOLO* berhasil mendeteksi keberadaan plat pada semua kondisi pengujian. Namun, hasil pembacaan teks nomor plat menggunakan *Tesseract OCR* masih menghasilkan output yang tidak sesuai dengan nomor plat asli.

Hasil ini menunjukkan bahwa integrasi *YOLOv8* dan *Tesseract OCR* masih menghadapi tantangan utama pada tahap *OCR*, bukan pada deteksi plat. *YOLO* telah berhasil menyediakan *bounding box* plat secara konsisten, tetapi *Tesseract* belum mampu mengenali karakter dengan akurat pada berbagai kondisi uji. Secara keseluruhan, hasil pengujian menunjukkan bahwa sistem sudah baik dalam deteksi plat, namun akurasi pembacaan nomor plat masih perlu ditingkatkan agar portal dapat terbuka secara otomatis sesuai validasi nomor plat yang terdaftar, sehingga tingkat keberhasilan dalam membuka pintu portal yang didapatkan adalah 0%.

Tabel 3. Hasil pengujian plat cetak putih DD 1356 MV

No.	Pengujian	Hasil Deteksi *YOLO*	Hasil OCR (*Tesseract*)	Status Portal
1	Siang, jarak 50 cm, sudut 15°	Plat terdeteksi	DD1356MV	Portal tidak terbuka
2	Siang, jarak 50 cm, sudut 30°	Plat terdeteksi	DD1356MV	Portal tidak terbuka
3	Siang, jarak 80 cm, sudut 15°	Plat terdeteksi	DD1356MV	Portal tidak terbuka
4	Siang, jarak 80 cm, sudut 30°	Plat terdeteksi	OD1356MV	Portal tidak terbuka
5	Malam, jarak 50 cm, sudut 15°	Plat terdeteksi	DD1356MV	Portal tidak terbuka
6	Malam, jarak 50 cm, sudut 30°	Plat terdeteksi	DD1356MV	Portal tidak terbuka
7	Malam, jarak 80 cm, sudut 15°	Plat terdeteksi	OD1356MV	Portal tidak terbuka
8	Malam, jarak 80 cm, sudut 30°	Plat terdeteksi	iD1356MV	Portal tidak terbuka

Pada pengujian kali ini dilakukan terhadap plat cetak putih dengan nomor DD 1356 MV pada kondisi siang dan malam, dengan variasi jarak 50 cm dan 80 cm, serta sudut kemiringan 15° dan 30°. Hasil pengujian mencakup tiga parameter utama, yaitu: keberhasilan deteksi plat menggunakan *YOLO*, hasil pembacaan teks menggunakan *Tesseract OCR*, dan status otomatisasi pembukaan portal. Secara keseluruhan, sistem deteksi berbasis *YOLO* menunjukkan kinerja yang baik karena pada seluruh skenario uji, objek plat berhasil terdeteksi. Hal ini menunjukkan bahwa algoritma *YOLO* mampu mengidentifikasi area plat nomor secara konsisten meskipun terdapat variasi jarak, sudut, dan kondisi pencahayaan (siang maupun malam). Dengan demikian, dapat disimpulkan bahwa faktor jarak dan sudut tidak signifikan memengaruhi proses deteksi awal *YOLO* terhadap keberadaan plat nomor, melainkan lebih berpengaruh pada tahap *OCR*.

Pada hasil *OCR* menunjukkan variasi tingkat akurasi dalam membaca teks plat. Pada jarak 50 cm, baik pada siang maupun malam, pembacaan OCR relatif akurat dengan hasil yang sesuai yaitu DD1356MV sehingga pintu portal dapat terbuka. Namun, pada jarak 80 cm dengan sudut 30° dan kondisi cahaya berbeda, terjadi beberapa kesalahan dalam mendeteksi nomor yang ada pada plat cetak putih dengan nomor DD 1356 MV, sehingga terdapat beberapa pengujian yang gagal dalam membuka pintu portal yang dikarenakan tidak sesuai dari data verifikasi dengan pembacaan *Tesseract OCR.* Tingkat keberhasilan yang di peroleh dari pengujian deteksi plat nomor cetak putih

adalah 62,5%. Adapun rumus yang digunakan dalam menentukan tingkat keberhasilan dalam pendeteksian plat nomor yaitu:

$$\frac{jumlah\ berhasil}{Total\ Percobaan} \times 100\%$$

4. KESIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan sistem pendeteksi nomor plat kendaraan berbasis *YOLOv8* dan *Tesseract OCR* yang dijalankan pada *Raspberry Pi* 4 untuk mengendalikan pintu portal otomatis. Kinerja sistem *YOLOv8* menunjukkan performa sangat baik dalam mendeteksi area plat nomor. *YOLOv8* mampu mengenali keberadaan plat dengan konsisten pada berbagai kondisi pencahayaan (siang dan malam), jarak (50–80 cm), serta sudut kemiringan (15°–30°). Faktor jarak dan sudut tidak terlalu memengaruhi proses deteksi awal. Tantangan utama terdapat pada tahap pengenalan karakter. Pada beberapa kondisi (jarak jauh atau sudut miring), *Tesseract OCR* masih menghasilkan kesalahan pembacaan huruf maupun angka. Misalnya, pada plat DP 2724 AY hampir seluruh hasil pembacaan tidak sesuai sehingga tingkat keberhasilan membuka portal adalah 0%, sementara pada plat DD 1356 MV akurasi lebih tinggi dengan tingkat keberhasilan 62,5%. Raspberry Pi 4, kamera USB, dan motor servo MG996R dapat bekerja secara sinkronndari proses pengambilan gambar hingga portal terbuka. Hal ini menunjukkan sistem cukup efisien untuk kebutuhan real-time pada skala kecil hingga menengah. Secara umum, penelitian ini membuktikan bahwa integrasi *YOLOv8* untuk deteksi objek dan *Tesseract OCR* untuk pembacaan teks dapat diterapkan pada perangkat *Raspberry Pi* 4 sebagai solusi sistem akses kendaraan otomatis. *YOLOv8* sangat efektif dalam deteksi plat nomor, namun akurasi *Tesseract OCR* masih perlu ditingkatkan agar sistem mampu membuka portal hanya untuk kendaraan terverifikasi secara lebih andal.

REFERENSI

Al-Hasan, T. M., Bonnefille, V., & Bensaali, F. (2024). Enhanced YOLOv8-Based System for Automatic Number Plate Recognition. *Technologies*, 12(9), Article 164. doi:10.3390/technologies12090164.

Al-Batat, R., Angelopoulou, A., Premkumar, S., Hemanth, J., & Kapetanios, E. (2022). An end-to-end automated license plate recognition system using YOLO-based vehicle and license plate detection with vehicle classification. *Sensors*, 22, 9477.

Juliano, J. J., Armada, M. K. A., Cosadio Jr., A. N. P., & Reyes, R. E. F. F. (2025). Automated Vehicle Access Control System Utilizing Computer Vision-Based License Plate Recognition. *International Journal of Multidisciplinary: Applied Business and Education Research*, 6(4), 1947–1977.

Moussaoui, H., El Akkad, N., Benslimane, M., El-Shafai, W., Baihan, A., Hewage, C., &

Rathore, R. S. (2024). Enhancing automated vehicle identification by integrating YOLO v8 and OCR techniques for high-precision license plate detection and recognition. *Scientific Reports*, 14, 14389. doi:10.1038/s41598-024-65272-1.

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 (CVPR)*, 779–788.

Smitha, B., Shivani, B., Sathish, D., Archana, N., & Rao, K. V. (2023). Vehicle Number Plate Recognition Using Raspberry Pi. *Journal on Electronic and Automation Engineering*, 2(1), 80–84.

Sonnara, F., Chihaoui, H., & Filali, F. (2025). Efficient real-time license plate recognition using deep learning on edge devices. *Journal of Real-Time Image Processing*, 22, 159. doi:10.1007/s11554-025-01738-3.

Thapliyal, T., Bhatt, S., Rawat, V., & Maurya, S. (2023). Automatic License Plate Recognition (ALPR) using YOLOv5 model and Tesseract OCR engine. In *Proceedings of the 2023 First International Conference on Advances in Electrical, Electronics and Computational Intelligence (ICAEECI)* (pp. 1–5). IEEE.

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. *Machine Learning and Knowledge Extraction*, 5, 1680–1716.